

**DISEÑO E IMPLEMENTACIÓN DE LA TELE-OPERACIÓN DE UN ROBOT
MANIPULADOR DEL LABORATORIO DE PROCESOS DE MANUFACTURA**

GIOVANNI ZORRILLA PRIETO

**UNIVERSIDAD AUTONOMA DE OCCIDENTE
DEPARTAMENTO DE AUTOMÁTICA Y ELECTRÓNICA
PROGRAMA DE INGENIERIA MECATRÓNICA
SANTIAGO DE CALI
2012**

**DISEÑO E IMPLEMENTACIÓN DE LA TELE-OPERACIÓN DE UN ROBOT
MANIPULADOR DEL LABORATORIO DE PROCESOS DE MANUFACTURA**

GIOVANNI ZORRILLA PRIETO

**Proyecto de grado para optar al título de
Ingeniero Mecatrónico**

**Director
JESUS ALFONSO LOPEZ SOTELO
Ingeniero Electricista
Magister en Automática
Doctor en Ingeniería**

**UNIVERSIDAD AUTONOMA DE OCCIDENTE
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA MECATRÓNICA
SANTIAGO DE CALI
2012**

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar al título de Ingeniero Mecatrónico.

JIMMY TOMBE ANDRADE

Jurado

JOSE FERNANDO GIL BOTERO

Jurado

Santiago de Cali, 17 de Agosto del 2012

AGRADECIMIENTOS

Agradezco primeramente a Dios por darme el tiempo de vida que llevo para culminar mis estudios de pregrado, por permitir y guiarme en las dificultades personales y profesionales que le han dado sabor a mis estudios, por proveer lo necesario para estudiar y por sobre todo el haberme permitido conocerlo.

Agradezco a mis padres por estar conmigo en las diferentes etapas de mi vida y ayudarme a alcanzar diversas metas, por ver mis sueños como los suyos y dar todo su esfuerzo y dedicación para educarme.

Agradezco a mi hermano Julián Andrés Zorrilla Prieto quien elaboró muchas de las imágenes que se encuentran a lo largo del documento.

CONTENIDO

	pág.
RESUMEN	13
INTRODUCCIÓN	14
1. PLANTEAMIENTO DEL PROBLEMA	16
2. ANTECEDENTES	17
3. JUSTIFICACIÓN	19
4. OBJETIVOS	21
4.1 OBJETIVO GENERAL	21
4.2 OBJETIVOS ESPECÍFICOS	21
5. MARCO TEÓRICO	22
5.1 EL ROBOT	22
5.2 TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES	28
5.3 TCP/IP	29
5.4 REAL-TIME TRANSPORT PROTOCOL	32
5.5 LENGUAJES DE PROGRAMACIÓN	35
5.6 LA COMUNICACIÓN EN SERIE	42
5.7 LABORATORIOS VIRTUALES Y REMOTOS EN LA EDUCACIÓN	51
6. DESARROLLO CONCEPTUAL	59

6.1	PROCESO DE IDENTIFICACIÓN DE NECESIDADES	59
6.2	ESTABLECER ESPECIFICACIONES DEL PRODUCTO	73
6.3	GENERACIÓN DE CONCEPTOS	85
6.4	SELECCIÓN DE CONCEPTOS	107
7.	DISEÑO A NIVEL DE SISTEMA	111
7.1	ARQUITECTURA DEL PRODUCTO	111
8.	PRUEBAS	120
9.	CONCLUSIONES	124
10.	RECOMENDACIONES	125
	BIBLIOGRAFIA	129

GLOSARIO

API: (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas¹.

APLICACIÓN: es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo. Esto lo diferencia principalmente de otros tipos de programas como los sistemas operativos (que hacen funcionar al ordenador), las utilidades (que realizan tareas de mantenimiento o de uso general), y los lenguajes de programación (con el cual se crean los programas informáticos)².

APPLET: es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador *web*. El *applet* debe ejecutarse en un contenedor, que lo proporciona un programa anfitrión, mediante un *plug-in*, o en aplicaciones como teléfonos móviles que soportan el modelo de programación por *applets*. No puede ejecutarse de manera independiente como lo hace una aplicación, ofrece información gráfica y a veces interactúa con el usuario, típicamente carece de sesión y tiene privilegios de seguridad restringidos. Normalmente lleva a cabo una función muy específica que carece de uso independiente. El término fue introducido en *AppleScript* en 1993³.

AUTÓMATA: del latín *automāta* y este del griego *automatos* (αὐτόματος), espontáneo o con movimiento propio. Según la RAE (Real Academia Española), máquina que imita la figura y los movimientos de un ser animado⁴.

¹ WIKIPEDIA: la enciclopedia libre. Interfaz de programación de aplicaciones [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 31 de enero del 2012 [Consultado el 22 de Marzo del 2012]. Disponible en internet:

<http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones>

² WIKIPEDIA: la enciclopedia libre. Aplicación informática [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 12 de marzo del 2012 [Consultado el 22 de Marzo del 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica>

³ WIKIPEDIA: la enciclopedia libre. Applet [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 19 de febrero del 2012 [Consultado el 19 de Marzo del 2012]. Disponible en internet: <<http://es.wikipedia.org/wiki/Applet>>

⁴ WIKIPEDIA: la enciclopedia libre. Autómata (mecánico) [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 01 de marzo del 2012 [Consultado el 19 de Marzo del 2012]. Disponible en

BACK-UP (Copia de seguridad): Es la copia total o parcial de información importante del disco duro, CDs, bases de datos u otro medio de almacenamiento. Esta copia de respaldo debe ser guardada en algún otro sistema de almacenamiento masivo, como ser discos duros, CDs, DVDs entre otros.

Los *backups* se utilizan para tener una o más copias de información considerada importante y así poder recuperarla en el caso de pérdida de la copia original⁵.

BIBLIOTECA: también conocida como librería, es un conjunto de subprogramas utilizados para desarrollar *software*. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Algunos programas ejecutables pueden ser a la vez programas independientes y bibliotecas, pero la mayoría de estas no son ejecutables. Ejecutables y bibliotecas hacen referencias (llamadas enlaces) entre sí a través de un proceso conocido como enlace, que por lo general es realizado por un *software* denominado enlazador⁶.

CRIBA DE ERATÓSTENES: es un algoritmo para encontrar los N primeros números primos. Se escriben los números desde el dos (2) hasta el número deseado (N) en un cuadro, después, se eliminan los números que sean múltiplos del dos (2) sin tocar a este, en seguida, se procede a encontrar y eliminar los múltiplos del siguiente número no eliminado, es decir, el tres (3). El siguiente número no eliminado es el cinco (5), aplicamos el procedimiento descrito, y así, vamos avanzando hasta el final. Finalmente habrán quedado solamente los números primos⁷.

GRADOS DE LIBERTAD (GDL): cada uno de los movimientos independientes que puede realizar cada articulación de un robot con respecto a la anterior. El número de GDL del robot viene dado por la suma de los GDL de las articulaciones

internet: <[http://es.wikipedia.org/wiki/Aut%C3%B3mata_\(mec%C3%A1nico\)](http://es.wikipedia.org/wiki/Aut%C3%B3mata_(mec%C3%A1nico))>

⁵ ALEGSA. Diccionario de Informática: Definición de Backup [en línea]. [Consultado el 19 de Marzo del 2012]. Disponible en internet: <<http://www.alegsa.com.ar/Dic/backup.php>>

⁶ WIKIPEDIA: la enciclopedia libre. Biblioteca (informática) [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 23 de febrero del 2012 [Consultado el 19 de Marzo del 2012]. Disponible en internet: <[http://es.wikipedia.org/wiki/Biblioteca_\(programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n))>

⁷ BARBERO CORRAL, Eduardo. Múltiplos y divisores: Números primos [en línea]. España: Instituto de Tecnologías Educativas (ITE) del Ministerio de Educación, Cultura y Deporte, 2004. [Consultado el 23 de Marzo del 2012]. Disponible en internet: <http://descartes.cnice.mec.es/materiales_didacticos/Multiplos_divisores/criba.htm>

que lo componen⁸.

GUARDAS DE SEGURIDAD: barreras físicas diseñadas como parte de, o agregadas a la máquina, que previenen el acceso a las zonas peligrosas de esta⁹.

INTERNET: (acrónimo del inglés *internetworking* – conexión de redes) es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California y una en Utah, Estados Unidos¹⁰.

MICROCONTROLADOR: es un circuito integrado o chip de alta escala de integración que normalmente dispone en su interior de los siguientes componentes: Procesador o CPU (Unidad Central de Procesamiento). Memoria RAM para contener los datos. Memoria para el programa tipo ROM/PROM/EPROM. Líneas de E/S para comunicarse con el exterior. Diversos módulos para el control de periféricos (temporizadores, Puertos Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc.). Generador de pulsos de reloj que sincronizan el funcionamiento de todo el sistema¹¹.

OPEN SOURCE: es *software* que está disponible en forma de código fuente: el código fuente y algunos otros derechos que normalmente se reserva para los titulares de derechos de autor se proporcionan bajo una licencia de *software* libre que permite a los usuarios estudiar, cambiar, mejorar y en ocasiones, también distribuir el *software*. Muy a menudo se desarrolla de una manera pública y

⁸ BARRIENTOS, Antonio; PEÑIN, Luís Felipe y BALAGUER, Carlos. Fundamentos de Robótica. McGraw Hill, 1996. p. 17.

⁹ GRANDA, Eduardo Javier. Guardas de Seguridad en Máquinas [en línea]. Argentina: Estrucplan Consultora S.A. Actualizado 27 de Octubre del 2011 [Consultado el 05 de Noviembre del 2011]. Disponible en internet: <<http://www.estrucplan.com.ar/articulos/verarticulo.asp?IDArticulo=2807>>

¹⁰ WIKIPEDIA: la enciclopedia libre. Open-source software [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 07 de marzo del 2012 [Consultado el 19 de Marzo del 2012]. Disponible en internet: <http://en.wikipedia.org/wiki/Open-source_software>

¹¹ BARRÓN ZAMBRANO, José Hugo y CERDA VILLAFANA, Gustavo. Manual de Microcontrolador 16F873 [en línea]. Guanajuato, México: Universidad de Guanajuato. p. 2. [Consultado el 19 de Marzo del 2012]. Disponible en internet: <<http://www.fimee.ugto.mx/profesores/gcerda/documentos/manual.pdf>>

colaborativa¹².

PLANTA: en teoría de Control, es un conjunto de partes que trabajan juntas con el objetivo de realizar una operación en particular. Se le llama planta al objeto físico que se va a controlar o sobre el que se efectúa el control¹³.

PLUG-IN: (del español "enchufable") también conocido como *add-on* (agregado), complemento, conector o extensión. Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. Los *plug-in* permiten que los desarrolladores externos colaboren con la aplicación principal extendiendo sus funciones para reducir el tamaño de la aplicación y/o separar el código fuente de la aplicación a causa de la incompatibilidad de las licencias de *software*¹⁴.

ROBOT MANIPULADOR: son sistemas mecánicos multifuncionales, formados generalmente por elementos en serie, articulados entre sí, destinados al agarre y desplazamiento de objetos; con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos¹⁵.

SISTEMA OPERATIVO: es un conjunto de programas o *software*, destinado a permitir la comunicación entre el usuario y la máquina de forma cómoda y eficiente; se encarga de gestionar los recursos del ordenador, esto incluye la gestión del *hardware* desde los niveles más básicos. Un sistema operativo se encarga de realizar funciones básicas para la gestión de un sistema informático, estas funciones son:

¹² WIKIPEDIA: la enciclopedia libre. Open-source software [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 07 de marzo del 2012 [Consultado el 19 de Marzo del 2012]. Disponible en internet: <http://en.wikipedia.org/wiki/Open-source_software>

¹³ NÚÑEZ ENRÍQUEZ, Fernando. Control de movimiento empleando Labview, un enfoque didáctico [en línea]. Trabajo de grado Licenciatura en Ingeniería en Electrónica y Computadoras. Puebla, México: Universidad de las Américas Puebla, 2007. 84 p. [Consultado el 19 de Marzo de 2012]. Disponible en internet: <http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/nunez_e_f/capitulo1.pdf>

¹⁴ WIKIPEDIA: la enciclopedia libre. Complemento (informática) [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 08 de Marzo del 2012 [Consultado el 19 de Marzo de 2012]. Disponible en internet: <<http://es.wikipedia.org/wiki/Plugin>>

¹⁵ BARRIENTOS, Op. Cit., p. 10.

Implementación interfaz de usuario. El medio por el cual interactúan usuario/sistema.

Administración de recursos. Gestiona el óptimo manejo de los recursos del sistema.

Administración de archivos. Controla la forma en que los archivos se almacenan física y lógicamente.

Administración de tareas. Gestiona los procesos que se ejecutan o que se ejecutarán en el procesador.

Servicio de soporte y utilidades. Asistencia y gestión de programas utilitarios del sistema¹⁶.

SLEW RATE: es un efecto no lineal en los amplificadores. Representa la incapacidad de un amplificador para seguir variaciones rápidas de la señal de entrada. También se le define como la velocidad de cambio del voltaje de salida con respecto a las variaciones en el voltaje de entrada. Se expresa típicamente en unidades de Voltio por microsegundo [$V/\mu s$]¹⁷.

TEACH PENDANT: es un dispositivo de uso opcional que permite controlar los movimientos del robot en coordenadas XYZ y por articulaciones. En adición, presenta la posibilidad de operar el robot a través de comandos¹⁸.

¹⁶ SOTO, Lauro. Definición Concepto Sistemas Operativos [en línea]. Ensenada, México: Mi tecnológico. [Consultado el 19 de Marzo de 2012]. Disponible en internet: <<http://www.mitecnologico.com/Main/DefinicionConceptoSistemasOperativos>>

¹⁷ OLIVEROS, Jeisson Leonardo. PARÁMETROS Y DIFERENCIAS DE ALGUNOS AMPLIFICADORES OPERACIONALES [en línea]. [Consultado el 19 de Marzo de 2012]. Disponible en internet: <http://omnisapiente.megatesis.com/index.php?option=com_content&task=view&id=1015&Itemid=308>

¹⁸ CASTILLO, Martín A. INFORME DESCRIPTIVO SCORBOT-ER Vplus: Robot de trayectoria continua [en línea]. Argentina: Universidad Nacional de Quilmes. p. 2. [Consultado el 19 de Marzo de 2012]. Disponible en internet: <iaci.unq.edu.ar/materias/c_robots/archivos/expo/informe_scorbot.doc>

TELEOPERAR: manipular u operar a distancia un dispositivo.

RESUMEN

En el presente proyecto se implementa un software para operar remotamente un robot manipulador, inicialmente se escoge el lenguaje de programación a usar teniendo en cuenta varias características de los lenguajes orientados a objetos, si es de escritura estática y el nivel Carpers Jones que determina la eficiencia del programa. En seguida, se escoge el robot más apropiado para trabajar en un Laboratorio Remoto, entre los robots disponibles en la Universidad Autónoma de Occidente.

Una vez escogido lo anterior, se determina las necesidades que presenta la universidad frente al Laboratorio de Procesos de Manufactura en lo relacionado a la operación de los manipuladores robóticos y la importancia relativa de los requerimientos del cliente.

Cuando se tiene claro lo que el usuario necesita se crean unas métricas que permitan medir las características que va a tener la solución del problema para así poder medir el grado de satisfacción del cliente ante el desarrollo del proyecto. Después, se determinan las funciones que componen el sistema para generar ideas que den respuesta a cada una de ellas. La suma y diferentes combinaciones de las ideas crean los posibles conceptos a implementar, su puesta en marcha depende de la satisfacción que traiga al cliente, por lo cual, se evalúan los conceptos para escoger la mejor opción.

Finalmente, se lleva a cabo el concepto escogido y se expone cada detalle de este.

Palabras claves: Manipulador Robótico, Lenguaje de Programación, Teleoperación, Laboratorio Remoto, TCP/IP, RTP, puerto serial, dirección ip, puerto TCP.

INTRODUCCIÓN

El mando a distancia o control remoto, normalmente se hace a través de un dispositivo electrónico, el cual manipula un objeto que se encuentra a distancia; esto con el objetivo de mejorar la seguridad, disminuir gastos, esfuerzos y tiempos de desplazamiento.

Uno de los primeros ejemplos de mando a distancia fue el “*telekino*”, hecho por Leonardo Torres Quevedo en 1903. Este autómata ejecutaba órdenes transmitidas mediante ondas hertzianas y constituyó el primer aparato de radio-dirección del mundo, haciendo del inventor un pionero en el campo del mando a distancia, junto a Nikola Tesla. En 1906, en presencia del Rey Alfonso XIII y ante una gran multitud, demostró con éxito el invento al guiar un bote desde la orilla en el puerto de Bilbao, España. Otro ejemplo, fue el primer dispositivo para controlar remotamente una televisión, desarrollado por Zenith Radio a principios de 1950's. El dispositivo llamado extraoficialmente “*Lazy Bones*” usaba un cable para conectarse al televisor. Para mejorar este sistema, en 1955 se creó un control remoto sin cables llamado “*Flashmatic*”, el cual funcionaba enviando un rayo de luz a una célula fotoeléctrica¹⁹.

En la actualidad, hay gran variedad de dispositivos controlados remotamente y de diferentes maneras. Una de las principales aplicaciones son los Laboratorios Virtuales y Remotos, los cuales han sido diseñados con el fin de que los estudiantes de cursos avanzados de Pre-grado, especialmente en el área de Control y Automática, puedan desarrollar sus prácticas de laboratorio sin la necesidad de estar en contacto físico con la planta o dispositivo, accediendo a ellos a través de un servidor *Web* conectado a la red, lo cual les permitirá desarrollar su práctica de laboratorio no solo en simulación (Laboratorio Virtual) sino también en tiempo real (Laboratorio Remoto), inclusive, en alguno casos es posible visualizar a través de una *webcam* la forma como se comporta físicamente la planta cuando se es accedida en modo remoto.

Es así como los laboratorios Virtuales y Remotos se han convertido en una herramienta de estudio útil en el campo de la Educación a distancia, revolucionando los sistemas de enseñanza en diferentes Universidades alrededor del mundo, y aunque las aplicaciones más habituales se centran en el campo de la

¹⁹ AUDIO Y VIDEO REVIEWS. Mando a distancia o control remoto [En línea]. Actualizado 14 de Marzo del 2011 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://audioyvideo.org/index.php?topic=2717.0>>

Ingeniería de Control, Automática y afines, ya se ha venido extendiendo sus aplicaciones a otros campos de estudio como la ciencia, física y matemática, sobre todo en el análisis y modelamiento de sistemas dinámicos, donde es posible analizar y/o predecir su comportamiento (teórico), causa – efecto.

1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la Universidad Autónoma de Occidente posee entre sus herramientas pedagógicas manipuladores robóticos, los cuales se encuentra en el laboratorio de Procesos de Manufactura de la universidad. Estos dispositivos se usan para prácticas presenciales dentro de la institución pero no se les da el uso esperado, debido a:

Disponibilidad del equipo. El auxiliar encargado del área tiene otras actividades laborales y no puede disponer del tiempo necesario para asegurarse de que los usuarios manipulen adecuadamente los brazos robóticos. Adicionalmente, la universidad tiene un horario de atención, el cual no se puede exceder, limitando la permanencia de la comunidad autónoma dentro de sus instalaciones.

Disponibilidad de los usuarios. Los usuarios pueden tener poco tiempo disponible debido al trabajo, carga académica, situaciones personales, entre otras; y el hecho de desplazarse hasta la universidad implica tiempo adicional y dinero.

Por otra parte, la educación ha venido experimentando cambios significativos en lo que a metodología y técnicas utilizadas para impartir el conocimiento se refiere, llevando a cabo una reestructuración conceptual y teórica, la cual se ve influenciada por las nuevas tecnologías de la información orientadas hacia fines educativos, donde el principal promotor de ello es la Internet. Es por ello, que se debe optar por mejorar las herramientas tecnológicas si se desea mantener el posicionamiento en la región.

Comparado con un laboratorio convencional, un laboratorio remoto puede operar un mayor número de horas y con una mayor cobertura territorial, gracias a Internet. Los alumnos pueden acceder a practicar con equipos modernos para entender su funcionamiento y capacidad, con lo cual salen mejor preparados para enfrentar su vida laboral. Adicionalmente, los laboratorios virtuales serían un apoyo ideal para los cursos a distancia que la universidad brinda y un motivo para crear nuevos cursos de estos.

Es por ello, que se va a diseñar e implementar un prototipo de teleoperación para uno de los robots manipuladores que se encuentran en la universidad, solucionando los inconvenientes actuales y presentando las ventajas que ello implica.

2. ANTECEDENTES

Hoy por hoy, diferentes instituciones educativas a nivel mundial han enfocado sus esfuerzos en lo que se conoce como educación a distancia, lo cual le ha permitido al estudiante ampliar y reforzar los conocimientos adquiridos en el aula de clase y ponerlos en práctica de forma eficaz y eficiente gracias a las herramientas tecnológicas ofrecidas por parte de la institución, en donde el acceso a los laboratorios, desde cualquier punto que se encuentre conectado a la red, le permite al estudiante interactuar en tiempo real con los dispositivos que se han dispuesto para ello, siendo estos aplicados principalmente en el área de Control y Automática.

Una de las universidades que es pionera en el diseño e implementación de Laboratorios Virtuales y Remotos, es la Universidad Nacional de Educación a Distancia (UNED), ubicada en la ciudad de Madrid, España. En el año 2007, culminó con el proceso de desarrollo e implementación de un Laboratorio Virtual y Remoto, el cual cuenta con tres de los modelos más comunes de un laboratorio destinado al estudio de la Ingeniería de Control y Automática, que le permiten al estudiante interactuar con ellos de una forma variada en tiempo real: sistema de tres tanques, sistema *heatflow* y el sistema motor eléctrico²⁰.

Estos sistemas fueron implementados con la opción de modo simulación o remoto usando la misma interfaz gráfica. Esto fue posible gracias a que la interfaz gráfica del cliente fue desarrollada completamente en *Easy Java Simulations* (EJS), una herramienta de simulaciones *open source* con la capacidad de poder generar potentes aplicaciones y Applets sin necesidad de tener conocimientos de programación avanzados.

En Colombia, finalizando el año 2007, la Universidad del Valle (UNIVALLE), Cali; y la Universidad del Quindío, Armenia; realizaron un trabajo conjunto para implementar un laboratorio distribuido con acceso remoto para la enseñanza de la robótica. El laboratorio permite el desarrollo de sesiones prácticas interactivas a estudiantes de ambas universidades sobre plataformas móviles (Pioneer 3DX) y sobre brazos robóticos (Mitsubishi RV-2AJ) usando el canal de comunicación de alta velocidad soportado por la Red Nacional Académica de Tecnología Avanzada RENATA. Las tareas normalmente desarrolladas involucran: programación,

²⁰ DEPARTAMENTO DE INFORMÁTICA Y AUTOMÁTICA. Laboratorios virtuales y remotos [en línea]. Madrid, España: Universidad Nacional de Educación a Distancia (UNED). [Consultado el 20 de Marzo de 2011]. Disponible en internet en: <<http://lab.dia.uned.es/rlab/index.html>>

supervisión y visualización usando video y/o una representación en 2D²¹.

En el 2009, un estudiante de la Universidad Autónoma de Occidente diseñó un Laboratorio remoto de control inteligente, y en seguida, en el año 2012, otro estudiante diseña la tele-operación de un robot móvil, todo esto con la idea de desarrollar toda una línea de laboratorios remotos para mejorar las herramientas educativas de la institución.

²¹ CAICEDO BRAVO, Eduardo., *et al.* Laboratorio Distribuido con Acceso Remoto para la Enseñanza de la Robótica [en línea]. Colombia. Revista Educación en Ingeniería. ISSN 1900-8260. Ed. Asociación Colombiana de Facultades de Ingeniería ACOFI. v.7 fasc.N/A. p 51-61, 2009. Disponible en internet: <<http://www.educacioneningeneria.org/index.php/edi/article/view/73/63>>

3. JUSTIFICACIÓN

El presente proyecto parte de la necesidad de darle el uso adecuado a las herramientas educativas disponibles en la universidad considerando que se ha invertido considerables sumas de dinero para mejorar el nivel académico de sus estudiantes adquiriendo herramientas como robots, plantas para control y automatización, entre otros dispositivos. Es de gran importancia, ampliar la relación costo-beneficio que presentan dichas herramientas y como respuesta a ello se propone la aplicación de las tecnologías para operación remota de dispositivos, como lo son Los Laboratorios Remotos y Virtuales basados en la Web. Los beneficios se presentan a continuación:

- El equipo se encuentra seguro frente a manipulación incorrecta, ya que no hay contacto directo con el robot y mediante programación se pueden evitar aquellas instrucciones perjudiciales para la máquina.
- Los usuarios pueden tener acceso para desarrollar sus prácticas desde cualquier lugar donde haya conexión a internet.
- Se pueden generar cursos virtuales alrededor de la máquina sobre control, automatización y robótica.
- Dado que el único recurso necesario en este tipo de laboratorio es el computador, los estudiantes podrán realizar simultáneamente el mismo experimento sin interferir con sus demás compañeros suprimiendo las limitaciones que representa tener muchos estudiantes matriculados en un curso y poco espacio físico para realizar los experimentos.
- El auxiliar a cargo de la máquina puede concentrarse en otras tareas al disminuir la atención que debe prestar al robot, tanto por manipulación como por establecimiento de turnos y horarios para las prácticas.
- El estudiante puede familiarizarse con el experimento previo a la clase presencial para enriquecer su aprendizaje.
- Permite comparar el comportamiento de modelos matemáticos frente a

dispositivos reales brindando una visión completa del fenómeno a estudiar.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Diseñar una plataforma que permita tele-operar un robot manipulador que se encuentra en el laboratorio de procesos de manufactura de la Universidad Autónoma de Occidente a través de la *web*.

4.2 OBJETIVOS ESPECÍFICOS

Seleccionar el robot manipulador a utilizar del laboratorio de procesos de manufactura.

Seleccionar un esquema adecuado para la tele-operación del robot manipulador.

Diseñar el entorno de trabajo para la tele-operación de robot manipulador.

Realizar pruebas de funcionamiento y conectividad entre el la aplicación remota y el robot manipulador.

5. MARCO TEÓRICO

5.1 EL ROBOT

El término robot procede de la palabra checa *robota*, que significa literalmente trabajo o labor. Tradicionalmente, *robota* era el periodo de trabajo que un siervo debía otorgar a su señor, alrededor de 6 meses al año. Fue empleado por primera vez en la obra teatral R.U.R. (Robots Universales de Rossum), estrenada en Enero de 1921 en Praga por el novelista y dramaturgo checo Karel Capek; aunque algunos autores le atribuyen el origen de esta palabra a Josef Capek, hermano del novelista. La obra fue un éxito inmediato y pronto se estrenó en multitud de teatros por toda Europa y Estados Unidos²².

La palabra que describe el campo de estudio asociado a los robots es la robótica, palabra que fue acuñada por el bioquímico, escritor y divulgador científico norteamericano de origen ruso Isaac Asimov en su corta novela *Runaround*, en el año 1942. La robótica concentra seis áreas de estudio: la mecánica, el control automático, la electrónica, la informática, y como ciencias básicas, la física y la matemática. Ahora bien, a pesar de que se tenga conocimiento del origen y significado de la palabra robot y de su área de estudio, existen ciertas dificultades a la hora de establecer una definición formal de lo que es un robot industrial como tal. La primera de ellas surge de la diferencia conceptual entre el mercado japonés y el euro americano de lo que es un robot y lo que es un manipulador. Así, mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación, el mercado occidental es más restrictivo, exigiendo una mayor complejidad, sobre todo en lo relativo al control.

En segundo lugar, y centrándose ya en el concepto occidental, aunque existe una idea común acerca de lo que es un robot industrial, no es fácil ponerse de acuerdo a la hora de establecer una definición formal. Además, la evolución de la robótica ha ido obligando a diferentes actualizaciones de su definición²³.

²² WIKIPEDIA: la enciclopedia libre. Karel Capek [En línea]. Florida: Wikimedia Foundation, 2006. Actualizado 10 de Febrero del 2011 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <http://es.wikipedia.org/wiki/Karel_%C4%8Capek>

²³ BARRIENTOS, Antonio; PEÑIN, Luís Felipe y BALAGUER, Carlos. Fundamentos de Robótica. McGraw Hill, 1996. p. 9-10.

La definición más comúnmente aceptada posiblemente sea la de la Asociación de Industrias Robóticas (RIA), según la cual: “un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas”²⁴. Esta definición, ligeramente modificada, ha sido adoptada por la Organización Internacional de Estándares (ISO del inglés *International Organization for Standardization*) que define al robot industrial como: “Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materias, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas”. Se incluye en esta definición la necesidad de que el robot tenga varios grados de libertad.

Existen aún más definiciones pero común en todas las anteriores es la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo. Un sistema robotizado, en cambio, es un concepto más amplio, engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente²⁵.

5.1.1 Clasificación del Robot Industrial. La maquinaria para la automatización rígida dio paso al robot con el desarrollo de controladores rápidos, basados en el microprocesador, así como un empleo de servos en bucle cerrado, que permiten establecer con exactitud la posición real de los elementos del robot y establecer el error con la posición deseada²⁶. Esta evolución ha dado origen a una serie de tipos de robots, que se citan a continuación:

5.1.1.1 Manipuladores. Son sistemas mecánicos multifuncionales, formados generalmente por elementos en serie, articulados entre sí, destinados al agarre y desplazamiento de objetos²⁷; con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, de los siguientes modos:

²⁴ ASOCIACIÓN DE INDUSTRIAS DE ROBÓTICA (RIA). Robot Términos y Definiciones [En línea]. Actualizado 19 de Enero del 2010 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://www.robotics.org/product-catalog-detail.cfm?productid=2953>>

²⁵ BARRIENTOS, Op. Cit., p. 11.

²⁶ GONZALEZ FERNANDEZ, Víctor. R. Robots Industriales [en línea]. Valladolid, España: Centros de Formación del Profesorado e Innovación Educativa (CFID), 2002. [Consultado el 20 de Marzo de 2011]. Disponible en internet: <http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm>

²⁷ BARRIENTOS, Op. Cit., p. 10.

- **Manual.** Cuando el operario controla directamente la tarea del manipulador.
- **De secuencia fija.** Cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.
- **De secuencia variable.** Se pueden alterar algunas características de los ciclos de trabajo²⁸.

5.1.1.2 Robots de repetición o aprendizaje. Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario en la fase de enseñanza, se vale de una pistola de programación con diversos pulsadores o teclas, *joysticks*, maniqués, o a veces, desplaza directamente la mano del robot. Los robots de aprendizaje son los más conocidos, hoy día, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de gestual²⁹.

5.1.1.3 Robots con control por computador. Son manipuladores o sistemas mecánicos multifuncionales, controlados por un computador, que habitualmente suele ser un microordenador. En este tipo de robots, el programador no necesita mover realmente el elemento de la máquina, cuando la prepara para realizar un trabajo. El control por computador dispone de un lenguaje específico, compuesto por varias instrucciones adaptadas al robot, con las que se puede confeccionar un programa de aplicación utilizando solo el terminal del computador, no el brazo. A esta programación se le denomina textual y se crea sin la intervención del manipulador³⁰.

5.1.1.4 Robots inteligentes. Son similares a los del grupo anterior, pero, además, son capaces de relacionarse con el mundo que les rodea a través de sensores y tomar decisiones en tiempo real (auto-programable). De momento, son muy poco conocidos en el mercado y se encuentran en fase experimental, en la que se esfuerzan los grupos investigadores por potenciarles y hacerles más efectivos, al mismo tiempo que más asequibles. La visión artificial, el sonido de

²⁸ GONZALEZ, Op. Cit., Disponible en internet: http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm

²⁹ Ibid., Disponible en internet: http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm

³⁰ Ibid., Disponible en internet: http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm

máquina y la inteligencia artificial, son las ciencias que más están estudiando para su aplicación en los robots inteligentes³¹.

5.1.1.5 Micro-robots. Con fines educativos, de entretenimiento o investigación, existen numerosos robots de formación o micro-robots a un precio muy asequible y, cuya estructura y funcionamiento son similares a los de aplicación industrial³².

Las anteriores no son las únicas maneras de clasificar los robots, cada una varía de acuerdo al criterio de clasificación y a la empresa que lo clasifica.

5.1.2 Estructura mecánica de un robot. Mecánicamente, un robot está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos. La constitución física de la mayor parte de los robots industriales guarda cierta similitud con la anatomía del brazo humano, por lo que en ocasiones, para hacer referencia a los distintos elementos que componen el robot, se usan términos como cuerpo, brazo, codo y muñeca.

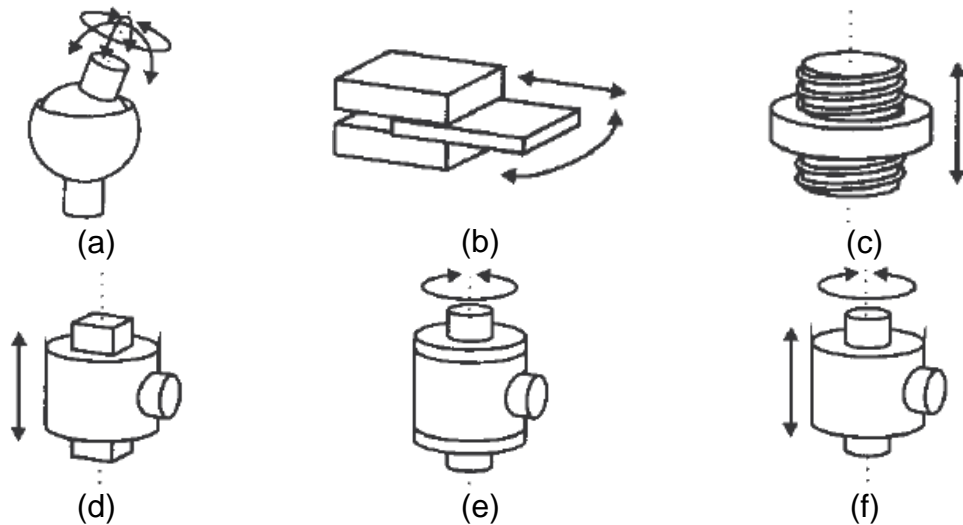
El movimiento de cada articulación puede ser de desplazamiento (traslación), de giro (rotación), o de una combinación de ambos. De este modo son posibles los seis tipos de articulación que se muestran en la figura 1, aunque, en la práctica, en los robots solo se emplean la de rotación y la prismática³³.

³¹ Ibíd., Disponible en internet: <http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm>

³² Ibíd., Disponible en internet: <http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm>

³³ BARRIENTOS, Op. Cit., p. 16.

Figura 1. Diferentes tipos de articulaciones. (a) Esférica o Rótula - 3 GDL. (b) Planar – 2 GDL. (c) Tornillo – 1 GDL. (d) Prismática – 1 GDL. (e) Rotación – 1 GDL. (f) Cilíndrica – 2 GDL



Fuente: BARRIENTOS, Antonio; PEÑIN, Luís Felipe y BALAGUER, Carlos. Fundamentos de Robótica. McGraw Hill, 1996. p. 17.

Cada uno de los movimientos independientes que puede realizar cada articulación con respecto a la anterior, se denomina Grado de Libertad (GDL). En la figura 1 se indica el número de GDL de cada tipo de articulación. El número de GDL del robot viene dado por la suma de los GDL de las articulaciones que lo componen. Puesto que, como se ha indicado, las articulaciones empleadas son únicamente las de rotación y prismática con un solo GDL cada una, el número de GDL del robot suele coincidir con el número de articulaciones de que se compone.

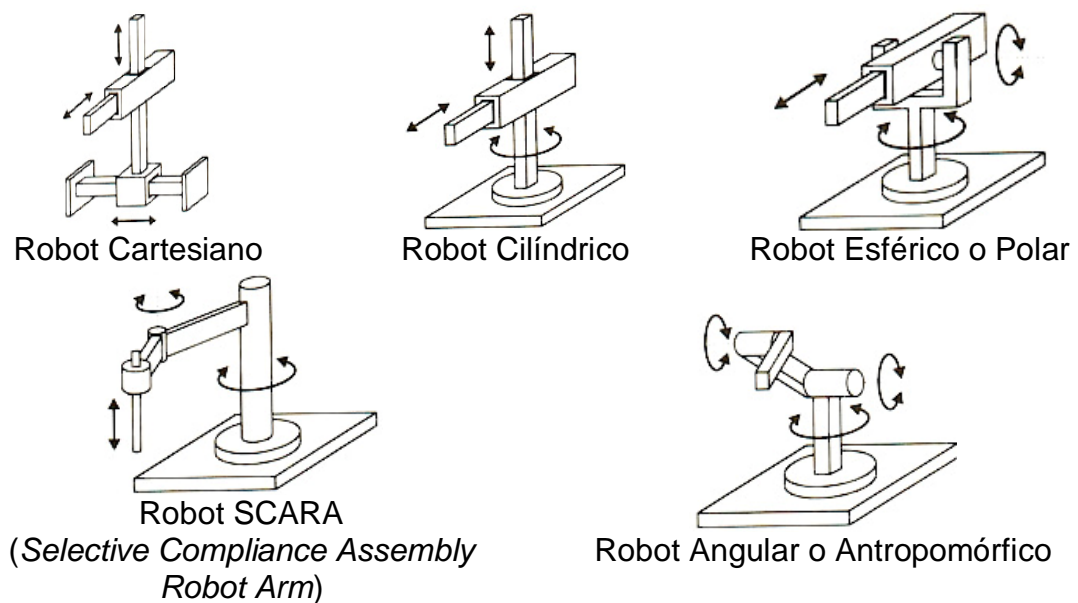
El empleo de diferentes combinaciones de articulaciones en un robot, da lugar a diferentes configuraciones, con características a tener en cuenta tanto en el diseño y construcción del robot como en su aplicación. Las combinaciones más frecuentes son las representadas en la figura 2 donde se atiende únicamente a las tres primeras articulaciones del robot, que son las más importantes a la hora de posicionar su extremo en un punto del espacio.

Puesto que para posicionar y orientar un cuerpo de cualquier manera en el espacio son necesarios seis parámetros, tres para definir la posición y tres para la orientación, si se pretende que un robot posicione y oriente su extremo de cualquier modo en el espacio, se precisaran al menos seis GDL.

En la práctica, a pesar de ser necesarios los seis GDL comentados para tener total libertad en el posicionado y orientación del extremo del robot, muchos robots industriales cuentan con solo cuatro o cinco GDL, por ser estos suficientes para llevar a cabo las tareas que se les encomiendan.

Existen también casos opuestos, en los que se precisan más de seis GDL para que el robot pueda tener acceso a todos los puntos de su entorno. Así, si se trabaja en un ambiente con obstáculos, al dotar al robot de grados de libertad adicionales le permitirá acceder a posiciones y orientaciones de su extremo a las que no hubiera llegado con seis GDL. Otra situación frecuente es la de dotar al robot con un GDL adicional que le permita desplazarse a lo largo de un carril, aumentando así el volumen del espacio que puede acceder. Cuando el número de grados de libertad del robot es mayor que los necesarios para realizar una determinada tarea se dice que el robot es redundante³⁴.

Figura 2. Configuraciones más frecuentes en robots industriales



Fuente: BARRIENTOS, Antonio; PEÑIN, Luís Felipe y BALAGUER, Carlos. Fundamentos de Robótica. McGraw Hill, 1996. p. 18.

³⁴ Ibíd., p. 17 y 19.

5.2 TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Las TIC conforman el conjunto de recursos necesarios para manipular la información, particularmente los ordenadores, programas informáticos y redes necesarias para convertirla, almacenarla, transmitirla y encontrarla. Las TIC se pueden segmentar de la siguiente manera:

- **Las redes.** Hacen referencia a los redes de acceso: telefonía, banda ancha, telefonía móvil, redes de televisión y redes en el hogar.
- **Los terminales.** Son los puntos de acceso, por lo tanto uno de los campos en los más evolución ha habido, aprovechando las nuevas infraestructuras y a la necesidad de poder intercambiar la información digital de mejor manera según lo deseen los usuarios: Ordenador personal, navegador de *Internet*, sistemas operativos para ordenadores, teléfono móvil, televisor, reproductores portátiles de audio y video, y consolas de juego.
- **Servicios.** Son los contenidos, servicios y aplicaciones a los que los usuarios tienen acceso y que están en constante innovación: Inicialmente estaban enfocados en la difusión de información estática, correo o buscadores, sin embargo, el aumento de ancho de banda y la sofisticación de las posibilidades tecnológicas y de evolución en la forma de prestar los servicios han permitido incrementar el portafolio disponible para los usuarios: Correo electrónico, búsqueda de información, banca en línea, audio y música, tv y cine, comercio electrónico, e-administración, e-gobierno, e-sanidad, educación, videojuegos, y servicios móviles.

En la educación las TIC han tenido un gran impacto, debido a que los estudiantes disponen de mayores medios para acceder a la información y a nuevas fuentes de información, permitiendo un aprendizaje más rápido e incluso de temas no relacionados a su currículo. También los centros educativos y culturales, como museos, bibliotecas y centros de recursos, emplean estos medios para difundir sus materiales y aumentar el material educativo disponible en la Internet, ayudando así a educar a una mayor cantidad de población.

La educación en los centros está adquiriendo mayor calidad y nivel de

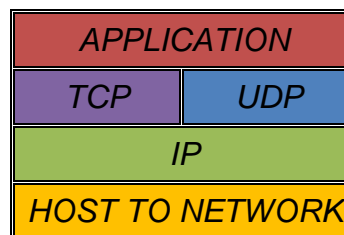
estandarización de los procesos educativos, pudiendo así ofrecer una formación base más profunda, más amplia y continua a lo largo de la vida de las personas³⁵.

5.3 TCP/IP

TCP/IP (*Transmission Control Protocol / Internet Protocol* en español Protocolo de Control de Transmisión / Protocolo de Internet) es el conjunto de protocolos utilizado por *Internet* (acrónimo de *Internetworking*). Puede trabajar sobre una multitud de protocolos como *Point to Point Protocol* (PPP), *FrameRelay*, X.25 o *Asynchronous Transfer Mode* (ATM)³⁶.

5.3.1 La pila de protocolos TCP/IP. Este no es un único par de protocolos, es un conjunto de ellos jerarquizado en distintas capas que conforman el denominado *STACK* o pila de protocolos TCP/IP. Posee sólo cuatro niveles, aunque en relación al Modelo OSI (*Open System Interconnection* en español Interconexión de Sistemas Abiertos) cubre la totalidad de capas del mismo. En la cuadro 1 se muestra el esquema del *stack* de protocolos.

Cuadro 1. El *stack* de protocolos TCP/IP



Fuente: LOPEZ FIGUEROLA, Mariano. Teoría de las redes informáticas [en línea]. Actualizado 21 de Julio del 2003 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://redesafull.galeon.com/#TCP/IP>>

³⁵ WIKIPEDIA: la enciclopedia libre. Tecnologías de la información y la comunicación [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 09 de Enero del 2012 [Consultado el 10 de Enero de 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Tecnolog%C3%ADas_de_la_informaci%C3%B3n_y_la_comunicaci%C3%B3n>

³⁶ LOPEZ FIGUEROLA, Mariano. Teoría de las redes informáticas [en línea]. Actualizado 21 de Julio del 2003 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://redesafull.galeon.com/#TCP/IP>>

- **Host to Network.** Abarca las capas Física y Enlace de OSI.
- **Internet Protocol (IP).** Equivale a la capa de Red de OSI.
- **Transport Control Protocol (TCP).** Corresponde a la capa de Transporte.
- **User Datagram Protocol (UDP).** También corresponde a la capa de Transporte.
- **Application.** Abarca las capas de Sesión, Presentación y Aplicación³⁷.

5.3.2 Internet Protocol (IP). A continuación se describen los puntos clave sobre el protocolo de internet IP.

5.3.2.1 Direccionamiento IP. La dirección IP es un conjunto de cuatro *bytes* denominados octetos, separados por puntos. Cada elemento conectado a una red TCP/IP debe tener una dirección IP única a fin de ser identificado en la misma en forma inequívoca. Las direcciones IP de los equipos se agrupan para poder identificar la red a la cual pertenece un determinado *Host* o equipo. Generalmente se expresan en forma decimal, lo cual significa que cada octeto puede ir desde 0 a 255, como por ejemplo, la dirección IP 192.234.15.122.

A fin de poder efectuar la agrupación antedicha, cada dirección IP se subdivide en dos partes: la primera parte identifica a la red y se denomina *NetID*. La segunda es la dirección del *host* o *HostID*.

No puede haber en una misma red o en *Internet* dos dispositivos conectados con una misma dirección IP, pero como hay equipos que se conectan a más de una red simultáneamente, un mismo equipo sí puede tener más de una³⁸.

³⁷ Ibíd., Disponible en internet: <<http://redesafull.galeon.com/#TCP/IP>>

³⁸ Ibíd., Disponible en internet: <<http://redesafull.galeon.com/#TCP/IP>>

5.3.3 Transmission Control Protocol (TCP). El protocolo TCP o Servicio de Transporte de Flujo Confiable es el encargado de asegurar un flujo de datos confiable entre los extremos de la red, por lo tanto, se encarga del control de flujo y del control de errores, tareas que no realiza IP. Trabaja en modo "*string*", es decir, recibe cadenas de bits de las capas superiores y las arma en segmentos que luego son enviados a la capa IP.

Aparece el concepto de Puerto, que es la entidad lógica que identifica los extremos de la comunicación, origen y destino. Los puertos se identifican por medio de un número de 16 *bits*, en otras palabras, un valor dentro del intervalo de cero (0) hasta 65535 ($2^{16} - 1$). Su función es identificar una aplicación a la cual se desea comunicar³⁹. Con el fin de unificar criterios en cuanto a los puertos que utilizarían las aplicaciones de *Internet*, la Autoridad de Números Asignados de *Internet* (del inglés *Internet Assigned Numbers Authority* – IANA) realizó una asignación de los números disponibles en tres categorías:

- **Puertos bien conocidos (del inglés *Well known ports*).** Comprendidos entre 0 y 1023. Estos 1024 puertos pueden ser representados con 10 *bits* y son usados por el sistema operativo o por procesos con privilegios para servicios conocidos, tales como, envío de correo electrónico, recepción de correo electrónico, navegar en *internet*, transferencia de archivos, entre otros. Las aplicaciones que usan este tipo de puertos son ejecutadas como servidores y se quedan a la escucha de conexiones.
- **Puertos registrados (del inglés *Registered ports*).** Son los puertos comprendidos entre 1024 y 49151. Son normalmente empleados por las aplicaciones de usuario de forma temporal cuando conectan con los servidores, pero también pueden representar servicios que hayan sido registrados por un tercero.
- **Puertos dinámicos y privados.** Son los puertos comprendidos entre los números 49152 y 65535, normalmente se asignan en forma dinámica a las aplicaciones de clientes al iniciarse la conexión⁴⁰.

³⁹ Ibíd., Disponible en internet: <<http://redesafull.galeon.com/#TCP/IP>>

⁴⁰ WIKIPEDIA: la enciclopedia libre. Puerto de red [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 07 de Diciembre del 2011 [Consultado el 10 de Enero del 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Puerto_de_red>

- Para elegir un puerto, se recomienda el uso de aquellos dinámicos y privados para evitar problemas de comunicación con alguna otra aplicación, adicionalmente, hay que tener en cuenta cuales puertos, en el intervalo recomendado de 49152 hasta 65535, son usados con frecuencia por aplicaciones mal intencionadas como los Troyanos (Ver anexo A).

De acuerdo al estado en que se encuentra un puerto se clasifican en: abierto, cerrado y sigiloso.

- **Abierto.** Acepta conexiones. Hay una aplicación escuchando en este puerto. Esto no quiere decir que se tenga acceso a la aplicación, sólo que hay posibilidad de conectarse.
- **Cerrado.** Se rechaza la conexión. Probablemente no hay aplicación escuchando en este puerto, o no se permite el acceso por alguna razón. Este es el comportamiento normal del sistema operativo.
- **Bloqueado o Sigiloso.** No hay respuesta. Este es el estado ideal para un cliente en *Internet*, de esta forma ni siquiera se sabe si el ordenador está conectado. Normalmente este comportamiento se debe a un corta-fuegos de algún tipo, o a que el ordenador está apagado⁴¹.

5.4 REAL-TIME TRANSPORT PROTOCOL

Tradicionalmente lo que ha interesado en la transmisión de datos por internet es que el receptor obtenga una copia exacta de los datos. HTTP, FTP, y en general todos los protocolos que funcionan sobre TCP priman la exactitud de la información y no la velocidad, en otras palabras, cuando un paquete se pierde o se daña, es retransmitido, por ello, la velocidad de transmisión se ralentiza por la sobrecarga de garantizar la entrega de los datos y además el orden en que llegan. Ahora bien, en aplicaciones en donde se requiere obtener la información en tiempo real, como en una videoconferencia o en la tele-operación de algún dispositivo, es muy importante la velocidad de presentación de los datos, por lo

⁴¹ SEGURIDAD INFORMÁTICA SMR. Tema 6: Seguridad en redes [en línea]. Actualizado 01 de Febrero de 2011 [Consultado el 10 de Enero del 2012]. Disponible en internet: <<http://seguridadinformaticasmr.wikispaces.com/TEMA+6+--+SEGURIDAD+EN+REDES>>

cual no es viable transmitir multimedia usando TCP, ya que, implicaría que la recepción presente retardos significativos de hasta varios segundos.

En su lugar se usan protocolos basados en User Datagram Protocol (UDP) que no es fiable, ya que, no garantiza que cada paquete llegue a su destino y en el orden en que fueron enviados, pero es más rápido. Para ello, el receptor tiene que ser capaz de compensar la pérdida de datos, los paquetes duplicados, y los paquetes que llegan fuera de orden. Lo anterior, no es un problema significativo en la transmisión multimedia, dado que, el usuario difícilmente percibe la pérdida de algun fotograma. Apoyándose en UDP se ha ideado un estándar de Internet para transportar datos en tiempo real, como audio y video, el Real-Time Transport Protocol (RTP) definido en Internet Engineering Task Force (IETF) RFC 1889.

RTP puede usarse como unicast o como multicast. Si se usa como unicast, copias separadas de los datos son entregadas a cada destinatario. Si usamos multicast los datos se envían desde el servidor solo una vez y la red es responsable de transmitir esos datos a múltiples destinos. El multicast es la forma más eficiente para muchas aplicaciones, como la videoconferencia⁴².

5.4.1 Servicios RTP

RTP permite identificar el tipo de datos que se está transmitiendo, determinar el orden en que los paquetes deben ser presentados y sincronizar la transmisión de diferentes fuentes.

Dado que RTP está basado en UDP, no proporciona un mecanismo para asegurar la entrega a tiempo u otras garantías de calidad del servicio, por ello, se acompaña de un protocolo de control, Real-Time Transport Control Protocol (RTCP) que permite monitorizar la calidad de la distribución de datos y proporciona mecanismos de control e identificación de las transmisiones RTP⁴³.

⁴² SUN MICROSYSTEMS. Java TM Media Framework API Guide [en línea]. 1999. p. 110-111. Disponible en internet: <http://java.coe.psu.ac.th/Extension/JMF/jmf2_0-guide.pdf>

⁴³ Ibíd., p. 111. Disponible en internet: <http://java.coe.psu.ac.th/Extension/JMF/jmf2_0-guide.pdf>

5.4.2 Arquitectura RTP

Una sesión RTP es una asociación entre un grupo de aplicaciones comunicándose con RTP. Una sesión se identifica por una dirección de red y un par de puertos. Un puerto se usa para transmitir los datos y el otro se usa para el control (RTCP).

Un participante es una máquina que, valga la redundancia, participa en la sesión. Participar en la sesión puede consistir, en la simple recepción pasiva de datos (receptor), en transmitir datos de forma activa (servidor), o ambos.

Cada medio (audio, video) se transmite en una sesión diferente. Por ejemplo, si en una conferencia se necesita audio y video, se usa una sesión para transmitir audio y otra para el video. Esto permite a los participantes elegir que medios quieren reproducir, por ejemplo, uno de los participantes con poco ancho de banda podría querer solo recibir el audio⁴⁴.

5.4.3 El protocolo de control RTCP

Cada participante en la sesión recibe periódicamente paquetes RTCP. Estos paquetes RTCP pueden contener información sobre la calidad del servicio, información sobre la fuente del contenido y estadísticas sobre los datos que se están transmitiendo.

Todos los participantes en la sesión mandan paquetes RTCP. Un participante que ha enviado paquetes de datos mandará un reporte del remitente (Sender Report - SR). El SR contiene el número total de paquetes y bytes enviados así como información que puede utilizarse para sincronizar las transmisiones de diferentes sesiones.

Los participantes en la sesión mandan periódicamente Reportes del Receptor (Receiver Report - RR) a todas las fuentes de las que están recibiendo datos. Un RR contiene información sobre el número de paquetes perdidos, el número de secuencia más alto recibido y una marca de tiempo que puede utilizarse para estimar el retardo de ida y vuelta entre el emisor y el receptor.

⁴⁴ Ibíd., p. 112. Disponible en internet: <http://java.coe.psu.ac.th/Extension/JMF/jmf2_0-guide.pdf>

El primer paquete en un paquete compuesto RTCP tiene que ser uno de reporte, incluso si no se han enviado o recibido datos, en cuyo caso se envía un RR vacío.

Todos los paquetes compuestos RTCP deben incluir una descripción del origen (Source DEscription - SDES) que contiene el nombre canónico (CNAME) que identifica el origen. Se puede añadir información adicional en el SDES como el nombre de la fuente, el email, número de teléfono, localización geográfica, nombre de la aplicación o un mensaje describiendo el estado actual del origen.

Cuando un origen se va a desconectar, manda un paquete BYE. El paquete BYE puede incluir el motivo por el que el participante abandona la sesión.

Los paquetes de aplicación proporcionan un mecanismo para que la aplicación defina y mande información a medida a través de este canal de control⁴⁵.

5.5 LENGUAJES DE PROGRAMACIÓN

Actualmente hay gran variedad de lenguajes de programación para diversos propósitos con sus respectivas fortalezas y debilidades, es por ello, que hay que escoger uno, si es necesario, para desarrollar la solución del proyecto. En esta sección se describe las características principales de los lenguajes de programación.

5.5.1 Paradigma de programación. Provee y determina la visión y métodos de un programador en la construcción de un programa o subprograma. Diferentes paradigmas resultan en diferentes estilos de programación y en diferentes formas de pensar la solución de problemas (con la solución de múltiples “problemas” se construye una aplicación)⁴⁶.

⁴⁵ Ibíd., p. 112-114. Disponible en internet: <http://java.coe.psu.ac.th/Extension/JMF/jmf2_0-guide.pdf>

⁴⁶ ALEGSA. Clasificación de los lenguajes de programación [en línea]. Actualizado 15 de Septiembre de 2006 [Consultado el 16 de Febrero de 2012]. Disponible en internet: <<http://www.alegsaonline.com/art/13.php>>

Los principales paradigmas son: Imperativo y Declarativo. De estos derivan otros paradigmas que son contados aparte por incluir nuevas características; estos se definen a lo largo de esta sección. Vale la pena aclarar que un paradigma no es mejor que otro, simplemente tienen ventajas y desventajas.

5.5.1.1 Imperativo. Se caracteriza por un modelo abstracto de ordenador que consiste en un gran almacenamiento de memoria. El ordenador almacena una representación codificada de un cálculo y ejecuta una secuencia de comandos que modifican el contenido de ese almacenamiento. Este paradigma viene bien representado por la arquitectura Von Neuman (1903-1957), ya que utiliza este modelo de máquina para conceptualizar las soluciones: "Existe un programa en memoria que se va ejecutando secuencialmente, y que toma unos datos de la memoria, efectúa unos cálculos y actualiza la memoria".

Si se analizan las características fundamentales de este tipo de paradigma se detectan las siguientes:

- **Concepto de celda de memoria ("variable") para almacenar valores.** Esta compuesto por un gran número de celdas donde se almacenan los datos. Las celdas tienen nombre (concepto de variable) que las referencian, y sobre los que se producen efectos de lado y definiciones de alias.
- **Operaciones de asignación.** Estrechamente ligado a la arquitectura de la memoria, se encuentra la idea de que cada valor calculado debe ser "almacenado", es decir asignado a una celda. Esta es la razón de la importancia de la sentencia de asignación en el paradigma imperativo. Las nociones de celda de memoria y asignación en bajo nivel, se tienden a todos los lenguajes de programación y fuerzan en los programadores un estilo de pensamiento basado en la arquitectura Von Neumann.
- **Repetición.** Realiza su tarea ejecutando repetidamente una secuencia de pasos elementales, ya que en este modelo computacional la única forma de ejecutar algo complejo es repitiendo una secuencia de instrucciones⁴⁷.

⁴⁷ WIKILEARNING: comunidades de wikis libres para aprender. Metodologías usadas en ingeniería del software - Paradigma imperativo [en línea]. Actualizado 25 de Octubre de 2007 [Consultado el 16 de Febrero de 2012]. Disponible en internet: http://www.wikilearning.com/curso_gratis/metodologias_usadas_en_ingenieria_del_software/3618-4

Dentro de esta categoría se encuentran el paradigma estructurado y el paradigma Orientado a Objetos.

- **Estructurado.** La programación se divide en bloques (procedimientos y funciones) que pueden o no comunicarse entre sí. Además la programación se controla con secuencia, selección e iteración. Permite reutilizar código programado y otorga una mejor comprensión de la programación⁴⁸. (Ver ejemplo en el anexo B).
- **Orientado a Objetos.** Este paradigma hace posible el desarrollo modular de un software, porque, hasta cierto grado, cada componente es independiente de los demás, lo que permite que estos componentes puedan ser reutilizados y reciclados, incluso, a lo largo de distintos proyectos. La definición exacta del término es muy variable dependiendo de a quién se le pregunte, pero hay varias cualidades que la mayoría están de acuerdo en que un lenguaje orientado a objetos debe tener:
 - **Objetos.** Están compuestos por estados y métodos. Los estados son propiedades del objeto y están representados por variables con valores únicos para cada objeto y que son llamadas variables de instancia. Por ejemplo, imagínese un perro, se podría pensar en cosas como: edad, color, tener hambre, tener ganas de salir al parque, etc., propiedades que son intrínsecas a cada perro pero cuyos valores son únicos en cada uno. Los métodos son representaciones de los comportamientos que el objeto es capaz de hacer. Por ejemplo, un perro en particular tendría los métodos: sentarse, echarse, comer, saludar, dormir y bailar, entre otros, y que son comportamientos que probablemente presentan otros perros.
 - **Encapsulación.** El código o datos de un objeto pueden estar ocultos para cualquier entidad externa a él.
 - **Herencia.** Es la propiedad de crear nuevos objetos a partir de la definición de otros. Un objeto “nuevo” será idéntico al modelo que seguimos para crearlo, excepto por algunos cambios incrementales o redefiniciones de sus estados o métodos.

⁴⁸ ALEGSA, Op. Cit., Disponible en internet: <<http://www.alegsaonline.com/art/13.php>>

- **Polimorfismo.** Esencialmente, es la propiedad que le permite a un método aceptar o devolver valores de más de un tipo. Formalmente se define así: siendo M un método y t1 y t2 son dos tipos de datos diferentes, M es polimórfico si M puede ser ejecutado con el argumento t1: M(t1) y el argumento t2: M(t2). De la misma manera una función puede ser definida como polimórfica si al llamarla mediante dos parámetros distintos, retorna valores de tipos diferentes.
- **Invocación Dinámica de Métodos (*Dynamic Method Binding*).** Cuando se llama a algún método de un objeto se ejecuta la operación específica de ese objeto y no alguna implementación de alguno de sus padres.
- **Todos los tipos predefinidos son objetos.**
- **Todas las operaciones son llevadas a cabo por el envío de mensajes a los objetos.** El mensaje es esencialmente una orden que se envía a un objeto para indicarle que realice alguna acción. También se define mensaje como la información completa que un objeto emisor transmite hacia un objeto receptor. Un mensaje contiene tres elementos: el objeto receptor, el tipo de evento y los datos necesarios para el receptor.
- **Todos los tipos definidos por el usuario son objetos.**

Para efectos del presente trabajo, un lenguaje es considerado como "puro" orientado a objetos si se cumplen todas estas cualidades. Un "híbrido" si puede soportar algunas de estas cualidades, pero no todas. En particular, muchos lenguajes soportan las tres primeras cualidades, pero no las últimas tres⁴⁹. (Ver ejemplo en el anexo B).

5.5.1.2 Declarativo. No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que se enfoca en describir las propiedades de la solución buscada, dejando indeterminado el algoritmo usado para encontrar esa solución. Es más complicado de implementar que el paradigma imperativo, tiene

⁴⁹ ZARATE REA, Héctor. Paradigmas de Programación [en línea]. Ciudad de México, México: Universidad Nacional Autónoma de México, 2008. p. 10-11. [Consultado el 21 de Febrero de 2012]. Disponible en internet: <<http://es.scribd.com/doc/9174723/Paradigmas-de-Programacion>>

desventajas en la eficiencia, pero ventajas en la solución de determinados problemas⁵⁰. Este paradigma se subdivide en el Lógico y el Funcional.

- **Funcional.** Está basado en el concepto matemático de función: una función f asigna a cada miembro de un conjunto X , exactamente un miembro de un conjunto Y , en donde el conjunto X y Y pueden o no ser el mismo, X es llamado dominio de f , y Y es llamado codominio o rango de f ; en otras palabras, considera al programa como una función matemática, donde el dominio representaría el conjunto de todas las entradas posibles y el rango sería el conjunto de todas las salidas posibles.

Una primera característica de los lenguajes funcionales es que no existe el concepto de variables y consecuentemente tampoco existen operaciones de asignación. Aunque podría parecer que los parámetros de una función se comportan como una variable, el valor de los parámetros es dado como fijo al inicio de la función. Otra característica es la existencia de una propiedad llamada transparencia referencial, la cual indica que una función sólo depende de sus parámetros y que tendrá efecto únicamente en su resultado, por lo que podríamos llamar a una función arbitrariamente sin tener efectos colaterales en el resto de las computaciones. Por último, una función puede ser utilizada como parámetros y resultados de cualquier otra función⁵¹. (Ver ejemplo en el anexo B).

- **Lógico.** Los programas están contruidos únicamente por expresiones lógicas, es decir, que son ciertas o falsas, en oposición a una expresión interrogativa (una pregunta) o expresiones imperativas (una orden). En este paradigma el orden de ejecución de las instrucciones no tiene nada que ver con el orden en que fueron escritas. Tampoco hay instrucciones de control propiamente dichas. (Ver ejemplo en el anexo B).

5.5.2 Tipado Estático y Dinámico. Los lenguajes de tipado (escritura) estático requieren que todas las variables se declaren con un tipo específico; por ejemplo, una cadena, un entero, booleano, punto flotante, etc. El compilador entonces se asegurará de que la variable contenga sólo un objeto compatible con ese tipo. Al imponer la restricción del tipo de objetos contenidos por la variable, el compilador puede garantizar un mensaje "no se entiende" cuando no hay compatibilidad. Por

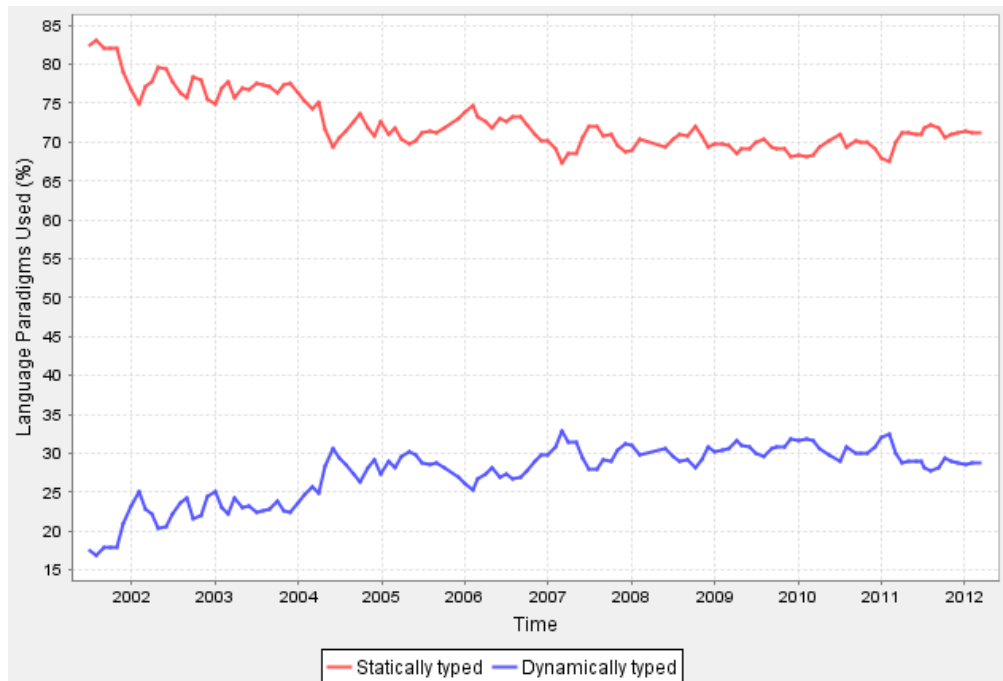
⁵⁰ ALEGSA, Op. Cit., Disponible en internet: <<http://www.alegsaonline.com/art/13.php>>

⁵¹ ZARATE, Op. Cit., p. 6-7. Disponible en internet: <<http://es.scribd.com/doc/9174723/Paradigmas-de-Programacion>>

otro lado, un sistema de tipado estático puede obstaculizar la evolución del software en algunas circunstancias, tal es el siguiente caso, si un método toma un objeto como parámetro, al cambiar el tipo del objeto se requiere cambiar la firma del método de modo que sea compatible con el nuevo tipo de objeto que se pasa. Si este mismo objeto se pasa a muchos métodos, todos ellos se deben actualizar en consecuencia, lo que podría ser una tarea ardua. Una ventaja es que puede marcar los errores en tiempo de compilación, señalando la ubicación exacta de los posibles errores.

Un sistema de tipado dinámico no requiere que las variables se declaren como un tipo específico, por ende, cualquier variable puede contener cualquier valor u objeto, lo cual presenta una desventaja, el usuario debe basarse en pruebas exhaustivas para asegurarse de que todos los usos indebidos de un objeto sean rastreados⁵². En la figura 3 se aprecia cual tipado prefieren los programadores según TIOBE.

Figura 3. Popularidad Lenguaje Tipo Dinámico versus Estático



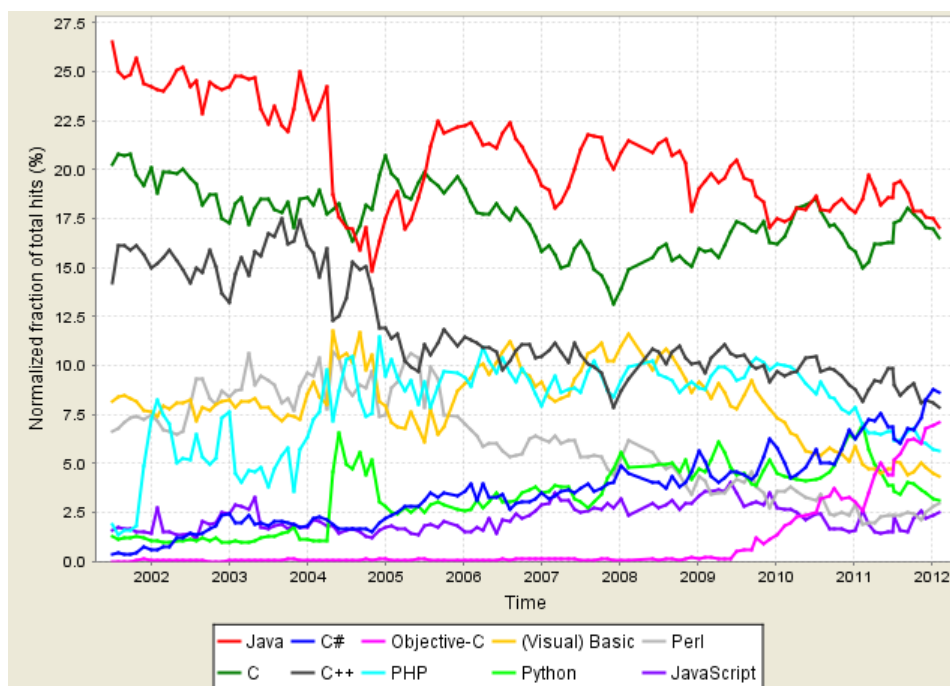
Fuente: TIOBE. TIOBE *Programming Community Index* [en línea]. Actualizado

⁵² VOEGELE, Jason. *Programming Language Comparison* [en línea]. [Consultado el 22 de Febrero de 2012]. Disponible en internet: <<http://www.jvoegele.com/software/langcomp.html>>

Marzo del 2012 [Consultado el 12 de Marzo de 2012]. Disponible en internet: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>

5.5.3 Popularidad. Este parámetro define cuan conocido o usado es un lenguaje de programación en relación a sus homólogos, en consecuencia, implícitamente ofrece la ventaja de tener un mantenimiento más fácil y económico. La economía de mantenimiento se abstrae teniendo en cuenta que la cantidad de mano de obra disponible es inversamente proporcional al valor cancelado por servicios de mantenimiento, es decir, cuando hay mayor cantidad de personas que puedan realizar el mantenimiento, menor será el valor que se estará dispuesto a pagar por este. La facilidad viene dada por la cantidad de personas que conocen del lenguaje de programación. En la figura 4 se ilustran los 10 lenguajes de programación más populares según TIOBE.

Figura 4. Popularidad de Lenguajes de Programación versus tiempo



Fuente: TIOBE. TIOBE *Programming Community Index* [en línea]. Actualizado Marzo del 2012 [Consultado el 12 de Marzo de 2012]. Disponible en internet: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>

5.5.4 Seguridad Integrada. Se refiere a la capacidad de implementación de un lenguaje para determinar si un trozo de código proviene de un origen de

"confianza" (tal como el disco duro del usuario), lo que limita los permisos del código si no es confiable. Por ejemplo, los *applets* de Java son considerados de confianza y, por lo tanto, están limitados en las acciones que pueden realizar cuando se ejecuta desde el navegador del usuario. No puede, por ejemplo, leer o escribir desde o hacia el disco duro del usuario, y no puede abrir una conexión de red a cualquier parte, sino al *host* de origen⁵³.

5.6 LA COMUNICACIÓN EN SERIE

La comunicación en serie es una forma de intercambiar información paso a paso, es decir, un dato a la vez. Esta forma de comunicarse no es nueva, es algo que se da a diario tanto en la comunicación entre diferentes dispositivos como entre personas. Un ejemplo de ello, es cuando se habla con alguien, en primer lugar se llama su atención y entonces se transmite el mensaje, una palabra cada vez. Cuando se termina, hay una pausa para indicar que hemos concluido. Lo mismo se cumple con la lectura o la escritura, se comienza una oración con la letra mayúscula, y se lee o escribe una palabra cada vez, con intervalos de cierto período⁵⁴.

En cuanto a dispositivos se refiere, en 1832, Samuel Morse inventó el telégrafo. Un dispositivo para la telecomunicación que enviaba dígitos alfanuméricos codificados en una serie de puntos y rayas. Esta codificación binaria es conocida como código Morse. El punto (marca) y la raya (espacio) en términos de computadoras serían equivalentes a uno (1) y cero (0), respectivamente.

5.6.1 Transmisión síncrona y asíncrona. La transmisión síncrona significa "al mismo tiempo" y la asíncrona, "en diferente tiempo". Ambas palabras vienen del griego y al descomponer "asíncrona" en su idioma origen, quedaría así, *a-syn-chronos* (*a* – "sin, carente de", *syn* – "con, juntamente, a la vez" y *chronos* – "tiempo").

Dado que, es imposible que un dispositivo receptor identifique por sí mismo cuando va a ser transmitida la información por el emisor, es necesaria una señal

⁵³ VOEGELE, Op. Cit., Disponible en internet: <<http://www.jvoegele.com/software/langcomp.html>>

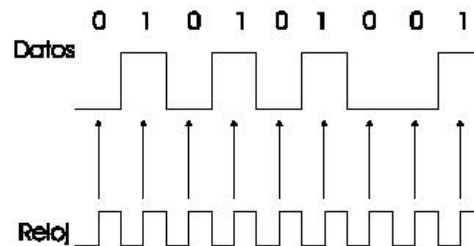
⁵⁴ TOBOSO, Emilio. La Comunicación Serie [en línea]. Actualizado 15 de Diciembre de 2010 [Consultado el 10 de Enero de 2012]. Disponible en internet: <<http://perso.wanadoo.es/pictob/comserie.htm>>

que le informe cuando va a comenzar la transmisión. Además, el dispositivo que recibe ignora cuán rápido tiene que leer los datos, debido a que, una máquina identifica las señales haciendo un muestreo de estas, es decir, cada cierto tiempo observa el estado de la señal, es por ello, que existe un problema básico en la identificación de la información (*bits* en este caso) que se transmite.

En la transmisión síncrona hay una señal denominada reloj que le indica al emisor con qué frecuencia envía cada *bit* y al receptor le da a conocer cuando hace el muestreo de la señal.

Si se utiliza el reloj de una de las dos máquinas (o un reloj tercero) se puede transmitir la señal de reloj por una línea auxiliar a la otra máquina⁵⁵. La figura 5 ilustra cómo a partir de la señal de reloj el muestreo es siempre exacto.

Figura 5. Datos y Señal de Reloj.



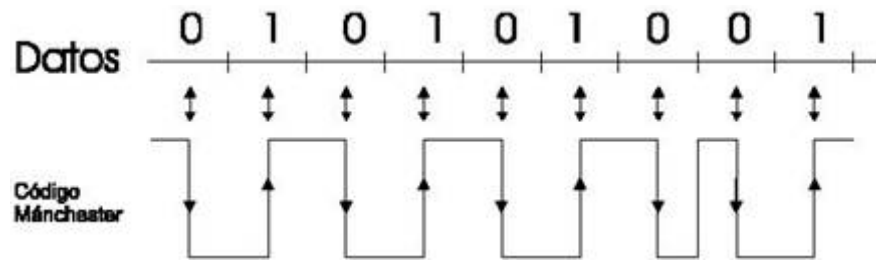
Fuente: LARA GOMEZ, Luis Felipe; DUQUE GIRALDO, Alexandra Milena y GARCIA SEPULVEDA, Juan Felipe. TX Serial. [Consultado el 10 de Enero de 2012]. Disponible en internet: <<http://pipelara20.tripod.com/txserial.htm>>

Otra forma, es que el emisor utilice una codificación para los datos, como por ejemplo el código Mánchester, en donde, hay un reloj entre los datos que marca el ritmo del muestreo que debe hacer el receptor. Se puede observar en la figura 6 que la codificación Mánchester tiene la propiedad de que existe siempre una transmisión en la mitad del intervalo del *bit*. El receptor aprovecha esta propiedad para sincronizarse en cada *bit*⁵⁶.

⁵⁵ LARA GOMEZ, Luis Felipe., DUQUE GIRALDO, Alexandra Milena., GARCIA SEPULVEDA, Juan Felipe. TX Serial. [Consultado el 10 de Enero de 2012]. Disponible en internet: <<http://pipelara20.tripod.com/txserial.htm>>

⁵⁶ Ibíd., Disponible en internet: <<http://pipelara20.tripod.com/txserial.htm>>

Figura 6. Datos en binario tradicional vs datos en código Mánchester.



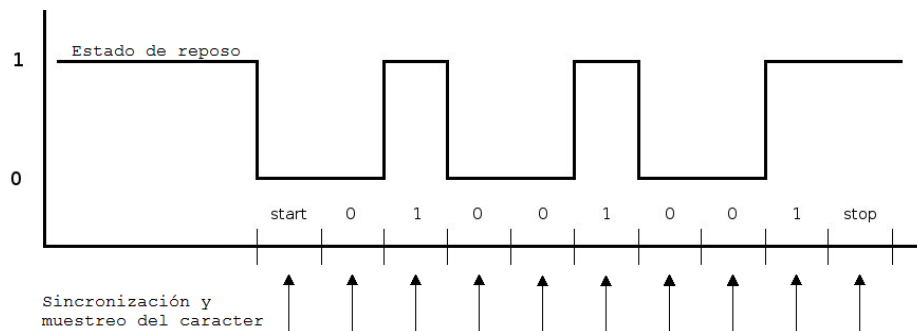
Fuente: LARA GOMEZ, Luis Felipe; DUQUE GIRALDO, Alexandra Milena y GARCIA SEPULVEDA, Juan Felipe. TX Serial. [Consultado el 10 de Enero de 2012]. Disponible en internet: <<http://pipelara20.tripod.com/txserial.htm>>

Con objetivo de mantener la sincronización, cuando no existen datos por enviar se transmiten caracteres sin valor.

En la transmisión asíncrona no existe la señal de reloj, pero esto se soluciona estableciendo a uno (1) lógico la salida del emisor cuando no transmite, luego, cuando decide transmitir un carácter, primero transmite un cero (0) que se denomina *bit* de *START* y sirve para que el receptor se sincronice, o en otras palabras, que empiece a contar los tiempos desde ese momento. El instante de sincronismo es el flanco de bajada de la señal (ver figura 7). Tras el *bit* de *START* se transmiten los bits de datos y después es obligatorio al menos un *bit* de *STOP* a uno (1) lógico⁵⁷. La secuencia se repite tantas veces como caracteres se transmitan. Hay que tener en cuenta que, ambas máquinas han sido previamente configuradas para emitir o recibir cierta cantidad de *bits*, de lo contrario, el receptor podría confundir el *bit* de *STOP* con un *bit* dato o viceversa.

⁵⁷ Ibíd., Disponible en internet: <<http://pipelara20.tripod.com/txserial.htm>>

Figura 7. Dato transmitido asincrónicamente



Fuente: LARA GOMEZ, Luis Felipe; DUQUE GIRALDO, Alexandra Milena y GARCIA SEPULVEDA, Juan Felipe. TX Serial. [Consultado el 10 de Enero de 2012]. Disponible en internet: <<http://pipelara20.tripod.com/txserial.htm>>; modificado por el autor.

5.6.2 Velocidad de Transmisión. Es expresada en *bits* por segundo (bps) o en baudios. El baudio es un concepto más general que *bit* por segundo. El baudio queda definido como el número de estados por segundo de la señal, si sólo existe dos estados (que pueden ser representados por un *bit*, que identifica dos unidades de información) entonces baudio es equivalente a *bit* por segundo. Baudio y *bit* por segundo se diferencian cuando es necesario más de un *bit* para representar más de dos estados de la señal⁵⁸.

5.6.3 Paridad. En el ambiente hay estática y diferentes señales de otros dispositivos electrónicos como celulares, televisores, radios, etcétera, los cuales, interfieren con los datos enviados cambiando uno o varios bits de datos de uno a cero o viceversa, a estas señales NO deseadas se les conoce como ruido. Conjuntamente, el cable por donde se transmite la señal consume potencia, alterando de otra forma cualquiera de los bits de datos. Para solucionar este problema existe la paridad. Este parámetro permite verificar la fidelidad de la señal recibida adicionando un bit al final del paquete de datos enviado. Existen cuatro diferentes tipos de paridad: par, impar, marcada y espaciada, así mismo, la opción de ninguna paridad también está disponible⁵⁹.

⁵⁸ TOBOSO, Op. Cit., Disponible en internet: <<http://perso.wanadoo.es/pictob/comserie.htm>>

⁵⁹ NATIONAL INSTRUMENTS. Comunicación Serial: Conceptos Generales [en línea]. Actualizado 02 de Enero del 2004 [Consultado el 15 de Diciembre del 2011]. Disponible en internet: <<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>>

5.6.4 Control de flujo. Es el proceso de gestión de la velocidad de transmisión de datos entre dos nodos para evitar que un dispositivo emisor transmita información a una velocidad mayor que el equipo de destino puede recibir y procesar. Esta medida proporciona un mecanismo al receptor para controlar la velocidad de transmisión, de modo que el receptor no se sobrecargue con los datos del transmisor. El control de flujo debe distinguirse del control de congestión, que se utiliza para controlar el flujo de datos cuando la congestión se ha producido⁶⁰. En lo que sigue, se describirán brevemente las cuatro formas más populares del control de flujo o *handshaking*: parada y espera, ventana deslizante, por software y por hardware.

5.6.4.1 Parada y espera. El método de parada y espera es para sistemas sincrónicos y funciona de la siguiente manera: cuando una computadora desea transmitir, envía un conjunto de bits (también llamada trama) que no puede exceder de una longitud determinada. Cuando la computadora destino recibe la información, envía una confirmación indicando que ha recibido la trama. La computadora de origen deberá esperar a recibir la confirmación, para poder continuar enviando más tramas. El nombre de esta técnica corresponde a la obligación de la computadora que emite, de esperar tras la emisión de una trama, hasta enviar la siguiente, o la misma en caso de una recepción errónea. Las ventajas de esta técnica son la relativa simplicidad de la misma. Por el contrario, se hace uso ineficiente del sistema, sobre todo en medios con elevado tiempo de propagación de las señales. La computadora esperará inútilmente la confirmación de la trama recibida, ya que en circunstancias normales de funcionamiento la recepción de datos será adecuada⁶¹.

5.6.4.2 Ventana deslizante. Este método también es para sistemas sincrónicos pero suple los aspectos negativos de la técnica anterior, ya que permite el envío de tramas mientras se espera la recepción de confirmación. El mecanismo es el siguiente: tanto el emisor como el receptor tiene un espacio igual donde almacenar una cantidad N de mensajes. La computadora origen puede enviar, uno tras otro, hasta la mitad de capacidad de mensajes de almacén, cada uno de los cuales ha de ir etiquetado con un número que sirve para obtener más tarde confirmación individual de cada trama. Mientras no llegue ninguna confirmación, el emisor no puede enviar más datos.

⁶⁰ WIKIPEDIA: la enciclopedia libre. Flow Control [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 17 de Septiembre de 2011 [Consultado el 15 de Diciembre del 2011]. Disponible en internet: <http://en.wikipedia.org/wiki/Flow_control>

⁶¹ WordPress.com. Mecanismos para la detección de errores [en línea]. Actualizado 11 de Septiembre del 2011 [Consultado el 15 de diciembre del 2011]. Disponible en internet: <<http://mediosdetransmisionyperturbaciones.wordpress.com/4-5/>>

El receptor debe confirmar cada trama según vaya llegando correctamente. De esta manera, mientras se envían las confirmaciones, el emisor puede estar mandando más información y, de esta forma se consigue un uso más apropiado de la conexión. Si las computadoras disponen de capacidad de almacenamiento de N mensajes, se dice que posee ventajas de tamaño N. La computadora receptora de mensajes va confirmando los mensajes en orden y ese espacio queda disponible para otros. En cierta medida los huecos libres se van desplazando para recibir nuevos mensajes, de ahí el nombre de esta técnica⁶².

5.6.4.3 Handshaking por software. Esta forma de sincronización utiliza *bytes* de datos como caracteres de control. Se usa en sistemas asincrónicos. Las líneas necesarias para la comunicación solamente son TxD, RxD, y GND, ya que los caracteres de control se envían a través de las líneas de transmisión como si fueran datos. Los caracteres XON y XOFF son enviados por el receptor para pausar al transmisor durante la comunicación.

A manera de ejemplo, asúmase que el transmisor comienza a enviar datos a alta velocidad. Durante la transmisión, el receptor se da cuenta que el búfer de entrada se está llenando debido a que la CPU está ocupado con otras tareas. Para pausar temporalmente la transmisión, el receptor envía XOFF (cuyo valor es típicamente 19 en decimal, o 13 en hexadecimal) hasta que el búfer se vacíe. Una vez que el receptor está preparado para recibir más datos envía XON (cuyo valor es típicamente 17 en decimal, u 11 en hexadecimal) para continuar la comunicación.

La mayor desventaja de este método es además lo más importante a considerar: los números decimales 17 y 19 son ahora los límites para la transmisión. Cuando se transmite en ASCII, esto no importa mucho ya que estos valores no representan carácter alguno. Sin embargo, si la transmisión de datos es en binario, lo más probable es que estos valores sean transmitidos como datos normales y falle la comunicación. Otra desventaja es que al usar la línea de transmisión para enviar caracteres de control compromete la velocidad con la que llegan los datos⁶³.

5.6.4.4 Handshaking por hardware. Este método utiliza líneas de hardware RTS/CTS y DTR/DSR que trabajan de manera conjunta siendo entrada/salida. El

⁶² Ibíd., Disponible en internet: <<http://mediosdetransmisionyperturbaciones.wordpress.com/4-5/>>

⁶³ National Instruments., Op. Cit., Disponible en internet: <<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>>

primer par de líneas es RTS (por sus siglas en inglés, *Request To Send*) y CTS (*Clear To Send*). Cuando el receptor está listo para recibir datos, cambia la línea RTS a estado alto; este valor será leído por el transmisor en la línea CTS, indicando que está libre para enviar datos. El siguiente par de líneas es DTR (por sus siglas en inglés, *Data Terminal Ready*) y DSR (*Data Set Ready*). Estas líneas se utilizan principalmente para comunicación por modem, permiten al puerto serial y modem indicarse mutuamente su estado. Por ejemplo, cuando el modem se encuentra preparado para que la PC envíe datos, cambia la línea DTR a estado alto indicando que se ha realizado una conexión por la línea de teléfono. Este valor se lee a través de la línea DSR y la PC comienza a enviar datos. Como regla general, las líneas DTR/DSR se utilizan para indicar que el sistema está listo para la comunicación, mientras que las líneas RTS/CTS se utilizan para paquetes individuales de datos⁶⁴.

5.6.5 Líneas o canales de comunicación. Se pueden establecer canales para la comunicación siempre tomando al microprocesador o micro-controlador como referencia (transmisor o TxD) y al periférico como destino (receptor o RxD), de acuerdo a tres técnicas: el *simplex*, el *duplex* y el *full duplex*.

5.6.5.1 Simplex. En este caso el emisor y el receptor están perfectamente definidos y la comunicación es unidireccional. Presentan la ventaja de tener un enlace con solo dos hilos pero la desventaja de que el extremo receptor no tiene ninguna forma de avisar al extremo transmisor sobre su estado y sobre la calidad de la información que se recibe. Por esta razón, generalmente no se utiliza⁶⁵.

5.6.5.2 Duplex, half duplex o semi-duplex. En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor en donde los datos se desplazan en ambos sentidos pero no de manera simultánea. Este tipo de comunicación se utiliza habitualmente en la interacción entre terminales y una computadora central. Su ventaja frente al *Simplex* es que permite la realización de procedimientos de detección y corrección de errores⁶⁶.

⁶⁴ Ibíd., Disponible en internet: <<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>>

⁶⁵ Wikipedia: la enciclopedia libre. Puerto Serie [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 14 de Noviembre de 2011 [Consultado el 10 de Enero de 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Puerto_serie>

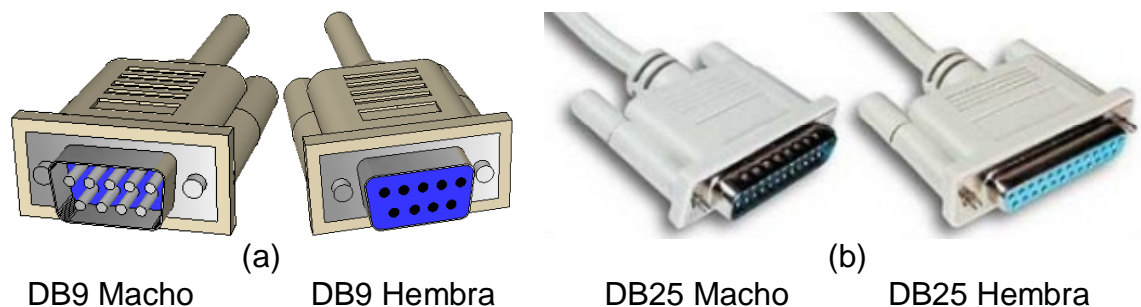
⁶⁶ Ibíd., Disponible en internet: <http://es.wikipedia.org/wiki/Puerto_serie>

5.6.5.3 Full Duplex. El sistema es similar al *Duplex*, pero los datos se desplazan en ambos sentidos simultáneamente. Para que sea posible ambos emisores poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados, mientras que la comunicación *semi-duplex* necesita normalmente uno solo. Para el intercambio de datos entre computadores este tipo de comunicaciones son más eficientes que las transmisiones *semi-duplex*⁶⁷.

5.6.6 Protocolo RS-232. Estándar Recomendado 232 (del inglés *Recommended Standard 232*) es un estándar publicado en 1969 por la EIA (*Electronic Standard Association*) definido para conectar un ordenador a un modem. El título de la norma dice exactamente: "Conexión entre un Equipo Terminal de Datos y un Equipo de Comunicación de Datos empleando un intercambio de datos binario serie". Por medio de este protocolo se estandarizan las características eléctricas y mecánicas de la conexión, además de la descripción funcional de las señales empleadas. Las características eléctricas están dispuestas en el anexo C.

En cuanto a las características mecánicas se refiere, la norma no establece el conector a usar, pero el más común es el conector de 25 pines (DB 25) o de 9 pines (DB 9), donde el conector macho identifica al DTE y el conector hembra al DCE. En la figura 8 están las imágenes de dichos conectores junto a una vista frontal de estos con la enumeración de sus pines.

Figura 8. Conectores RS232



Fuente: (b) DICORFIN IMPORTACIONES S.L. Cables & Adaptadores: Varios [en línea]. [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://www.dicorfin.com/default.asp?ididioma=2&idSeccion=productos-listado&id1=50&aux=Varios>; modificado por el autor.

⁶⁷ Ibid., Disponible en internet: http://es.wikipedia.org/wiki/Puerto_serie

En la descripción funcional de las señales usadas, las señales están básicamente divididas en seis categorías:

- **Señales de tierra y protección.**
- **Canal primario de comunicaciones.** Este se utiliza para el intercambio de datos, e incluye las señales de control de flujo.
- **Canal secundario de comunicaciones.** Cuando se implementa, se utiliza para el control del módem remoto, las solicitudes de retransmisión cuando se producen errores, y el gobierno sobre la configuración del canal principal.
- **El estado del módem y las señales de control.** Estas señales indican el estado del módem y proporcionan puntos de control intermedios, como el canal de voz de teléfono está establecido.
- **Señales de tiempo del transmisor y el receptor.** Si un protocolo síncrono se utiliza, estas señales proporcionan información de sincronización para el transmisor y el receptor, que puede funcionar a diferentes velocidades.
- **Las señales del canal de prueba.** Antes de que se intercambien datos, el canal puede ser probado por su integridad, y la velocidad de transmisión se ajusta automáticamente a la velocidad máxima que el canal soporta⁶⁸.

En el anexo D se enumeran los pines del DB25 con su respectivo nombre y función. El contenido del cuadro 2 es una breve descripción de los pines del DB9. No se entra en detalle puesto que las funciones han sido descritas anteriormente en el conector DB25.

⁶⁸ CAMI Research Inc. The RS232 Standard: Signal Definitions [en línea]. [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>

Cuadro 2. Descripción funcional de los pines del DB9 macho

Pin	Entrada/ Salida	Nombre Señal	Descripción
1	Entrada	DCD	Data Carrier Detect
2	Entrada	RxD	Receive Data
3	Salida	TxD	Transmit Data
4	Salida	DTR	Data Terminal Ready
5	-	GND	Ground
6	Entrada	DSR	Data Set Ready
7	Salida	RTS	Request To Send
8	Entrada	CTS	Clear To Send
9	Entrada	RI	Ring Indicator

Fuente: ZYTRAX INC. RS-232 Cables, *Wiring and Pinouts: RS232 on DB9 (EIA/TIA 574)* [en línea]. [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://www.zytrax.com/tech/layer_1/cables/tech_rs232.htm>; modificado por el autor.

A nivel de *software*, la configuración principal que se debe dar a una conexión a través de puertos seriales RS-232 es básicamente la selección de la velocidad en baudios (75, 110, 150, 300, 600, 1200, 2400, 4800, 9600 o 19200), la paridad, los bits de parada, la cantidad de bits por dato, que se utiliza para cada símbolo o carácter enviado y el control de flujo (Xon/Xoff, *hardware* o ninguno), donde su uso es opcional dependiendo del dispositivo a conectar.

Cada carácter se envía en una armazón (*frame*) consistiendo del *bit de START*, seguido por el caracter mismo, luego (opcionalmente) por el *bit* de paridad, y después el *bit de STOP*⁶⁹. El nombre caracter se usa porque cada grupo de *bits* representan una letra del alfabeto cuando el texto está codificado en ASCII.

5.7 LABORATORIOS VIRTUALES Y REMOTOS EN LA EDUCACIÓN

La aplicación del concepto nuevas tecnologías en la enseñanza en lo que

⁶⁹ VELEZ LOPEZ, Juan David y RIOS, Arcesio. Transmisión Serial [en línea]. Medellín, Colombia: Politécnico Colombiano Jaime Isaza Cadavid. [Consultado el 19 de Enero del 2012]. Disponible en internet: <<http://redesvelez.tripod.com/tserial.htm>>

conciernen a la realización de prácticas ha dado lugar a la aparición de diferentes modalidades de entornos de experimentación. Desde el punto de vista del estudiante/usuario, los criterios que permiten establecer una clasificación muy clara de estos nuevos entornos son dos: la forma de acceder a los recursos sobre los que se experimenta y la naturaleza del sistema sobre el que se opera. Atendiendo al primer criterio, se puede discernir entre acceso remoto a través de una red y acceso local. En cuanto a la naturaleza del recurso se refiere, hay que distinguir entre recurrir a modelos simulados o trabajar con plantas reales. De la combinación de estos dos criterios se obtienen cuatro clases de entornos muy diferentes, pero que abarcan todas las formas de experimentación posibles:

- **Acceso local - recurso real.** Representa el laboratorio de prácticas tal y como lo conocemos, en este el alumno se sitúa frente a un ordenador conectado a un sistema real para proceder a la realización de la práctica correspondiente.
- **Acceso local - recurso simulado.** Todo el entorno de trabajo es *software* y la interfaz de experimentación opera sobre un sistema simulado, virtual y físicamente inexistente que reside en el mismo ordenador en el que se encuentra la interfaz.
- **Acceso remoto - recurso real.** Constituye el acceso de un laboratorio real a través de una red. El usuario opera y controla de forma remota sistemas reales mediante una interfaz de experimentación que se ejecuta en un ordenador conectado a una red. Este enfoque es lo que se denomina tele-laboratorio, laboratorio remoto o tele-operación a través de la *web*.
- **Acceso remoto - recurso simulado.** Constituye el acceso de un laboratorio a través de una red pero el sistema real se sustituye por un modelo, por lo que el estudiante trabaja con su interfaz de experimentación sobre un sistema virtual accesible a través de *Internet*. Presenta como diferencia que pueden trabajar múltiples usuarios simultáneamente sobre el mismo sistema virtual ya que al estar simulado se puede instanciar para atender a todo aquel que lo solicite. Estamos pues ante la figura del laboratorio virtual multiusuario o simulación basada en el *web* - SBW (WBS por sus siglas en inglés de *web-based simulation*).

De las dos formas de acceder a los recursos, local o remota, la que tiene mayor demanda en la actualidad es esta última: el acceso a los recursos de experimentación a través de la *World Wide Web* (www). Esto ha dado lugar a la

aparición del término laboratorios basados en el web (*web-based* laboratorios) que pueden dividirse en laboratorios virtuales y laboratorios remotos según sea el tipo de recurso sobre el que se opera⁷⁰.

5.7.1 Laboratorios Virtuales. Pese a que la definición de laboratorio virtual puede parecer bastante intuitiva, es necesario profundizar en ella realizando una clasificación de las diferentes formas de diseñar un laboratorio virtual de acuerdo a cuatro criterios:

- La ubicación del motor matemático de cálculo.
- La naturaleza del núcleo de simulación.
- Las capacidades de diseño que tiene disponibles el usuario.
- El grado de interactividad con la simulación.

Acorde con el primer criterio, la ubicación del motor de cálculo, se puede distinguir entre local o remota. Dentro del ámbito en que nos movemos, las simulaciones locales se caracterizan porque el motor de cálculo se encuentra en el computador en el que está trabajando el usuario, de forma que la interfaz gráfica y el núcleo numérico forman un todo que convive en el mismo entorno, es decir, dentro del navegador.

Por el contrario, en las simulaciones remotas el núcleo numérico se ejecuta en un ordenador remoto, no siendo necesariamente en aquel desde el que se ha descargado la página HTML que alberga la interfaz gráfica, aunque lo habitual es que sí lo sea. En cualquier caso, la interfaz gráfica y la simulación se ejecutan en ordenadores diferentes.

El segundo criterio de clasificación es la naturaleza del núcleo de simulación con independencia de que su ubicación sea local o remota. Este criterio considera si la

⁷⁰ Departamento de Informática y Automática., Op. Cit. Disponible en internet: <<http://lab.dia.uned.es/rlab/index.html>>

simulación en sí misma ha sido construida por medio de una herramienta específica orientada al modelado y la simulación (MatLab, Simulink, ACSL, Dymola, Ecosim, etc.), o se ha recurrido a lenguajes de alto nivel de propósito general (C, C++, Fortran, Java) mediante el empleo de librerías específicas orientadas a la simulación.

Las capacidades de diseño consideran que el cliente pueda cambiar no sólo los parámetros numéricos del modelo a simular, sino que tenga la posibilidad de modificar su arquitectura. De esta forma, el cliente no se limita únicamente a la introducción de parámetros para configurar el comportamiento del modelo sino que toma parte activa en la construcción del propio modelo.

La cuarta y última característica que distingue las diferentes aproximaciones que existen para la realización de simulaciones basadas en el *web* es el grado de interactividad con la simulación. Podemos distinguir dos casos: *pseudo-batch* y *on-line*. En la simulación *pseudo-batch* no hay inmediatez desde que se inicia el proceso de simulación hasta que se obtiene la respuesta en forma de datos numéricos o gráficos. En algunos casos la respuesta del sistema puede corresponder a pasos intermedios del proceso de simulación. Por lo tanto, es posible introducir ciertos parámetros y repetir alguno de los pasos intermedios.

La simulación *on-line* representa el polo opuesto pero, a su vez, es el más sugerente. En este caso, el proceso de simulación avanza de forma continua y dinámica, obteniendo el usuario en cada periodo de muestreo de tiempo simulado los resultados bajo la forma de un flujo continuo de valores numéricos o de gráficos evolucionando de forma sostenida. Otra gran diferencia con respecto al supuesto previo es que, según se modifica un parámetro en la interfaz, la respuesta del sistema es inmediata (evidentemente, la rapidez de la reacción depende de la magnitud del cambio y de la dinámica del modelo)⁷¹.

Resumiendo, sin tener en cuenta las posibilidades de diseño y el grado de interactividad, los laboratorios virtuales más frecuentes son tres:

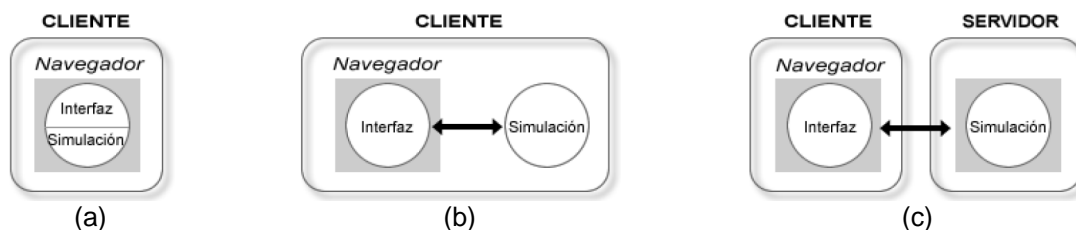
- **Laboratorio Virtual Monolítico.** Se descarga una aplicación que se ejecuta dentro del navegador web y reside en el computador del cliente. Es decir, la interfaz y el motor de simulación constituyen un objeto único. No se necesita la

⁷¹ Ibíd., Disponible en internet: <<http://lab.dia.uned.es/rlab/index.html>>

instalación de ningún entorno de simulación, salvo los correspondientes *plugins*, *run-time* de Java o LabVIEW (Ver figura 9a).

- **Laboratorio Virtual Híbrido.** Es análogo al monolítico, la interfaz gráfica y el motor son aplicaciones independientes pero necesita obligatoriamente que el cliente las tenga instaladas en su computador. El motor normalmente es un entorno de simulación del tipo Matlab/Simulink mientras que la interfaz se localiza dentro del navegador *www* (Ver figura 9b).
- **Laboratorio Virtual Distribuido.** El cliente ejecuta la interfaz en el navegador *web* mientras que el motor numérico es una aplicación (por ejemplo Matlab y Simulink) residente del lado del servidor, es decir, la interfaz gráfica y el motor son aplicaciones independientes y están físicamente separadas (Ver figura 9c). El dialogo se establece mediante un servidor de manejo de solicitudes / respuestas usando alguna tecnología *web* dinámica del lado del servidor (CGI, PHP, ASP, JSP, o *JavaServlets*) o del lado del cliente (*JavaScript*, *Applets* de Java o *ActiveX*)⁷².

Figura 9. Configuraciones más habituales de SBW: (a) Laboratorio Virtual Monolítico, (b) híbrido y (c) distribuido



Fuente: TORRES, Leticia Esmeralda. Simulación de Sistemas de Control en JAVA: Bases Conceptuales y Metodológicas para un Laboratorio Virtual [en línea]. Trabajo de grado Ingeniero de Sistemas. Mérida, Venezuela: Universidad de los Andes, 2006. p. 9. [Consultado el 19 de Agosto del 2011]. Disponible en internet: <http://tesis.ula.ve/pregrado/tde_busca/archivo.php?codArchivo=728>

⁷² TORRES, Leticia Esmeralda. Simulación de Sistemas de Control en JAVA: Bases Conceptuales y Metodológicas para un Laboratorio Virtual. Trabajo de grado Ingeniero de Sistemas. Mérida, Venezuela: Universidad de los Andes, 2006. p. 9-10. [Consultado el 20 de Marzo de 2011]. Disponible en internet: <http://tesis.ula.ve/pregrado/tde_busca/archivo.php?codArchivo=728>

5.7.1.1 Ventajas de los Laboratorios Virtuales. Los laboratorios virtuales implican muchos beneficios gracias a que las diferentes prácticas no ocupan un espacio físico y a que se puede tener acceso a ellas desde un computador, las ventajas se mencionan a continuación:

- **Costo.** Debido a que los computadores reemplazan la mayoría de los equipos del laboratorio, los costos de adquisición, almacenamiento y mantenimiento se ven reducidos, sin embargo esto se encuentra sujeto a los costos involucrados en el desarrollo del *software* requerido.
- **Seguridad.** Un laboratorio virtual puede filtrar la configuración de parámetros potencialmente destructivos en la vida real, o simular el daño que podrían causar.
- **Disponibilidad espacial y temporal.** Si un laboratorio se encuentra disponible a través de la red, podemos acceder a este en cualquier momento y desde cualquier lugar.
- **Familiarizarse con el experimento.** Evitando que los estudiantes puedan acudir al aula sin haber realizado trabajo previo.
- **Optimizar el uso de los recursos.** Los estudiantes requieren menos tiempo para realizar las prácticas haciéndose un mejor uso de los laboratorios locales y remotos.
- **Disminución del uso incorrecto del equipo.** Frecuentemente los dispositivos utilizados en los laboratorios reales son delicados, lo cual se acentúa si se les coloca a operar fuera de las condiciones de trabajo para las cuales fueron diseñados.
- **Comparación del comportamiento de modelos matemáticos frente a dispositivos reales.** Los modelos matemáticos se obtienen simplificando el comportamiento de los dispositivos reales, lo cual en algunos casos puede ocasionar un comportamiento diferente entre ambos.
- **Repetitividad de los experimentos.** Dado que el comportamiento de los

sistemas a estudiar se obtiene mediante el modelado matemático de la realidad, los alumnos pueden repetir de forma inequívoca las condiciones bajo las que se realizaron los experimentos, presentando siempre el mismo resultado.

- **Multiplicidad de experimentos simultáneos.** Debido a que el único recurso necesario en este tipo de laboratorio es el computador, los estudiantes podrán realizar simultáneamente el mismo experimento sin interferir con sus demás compañeros⁷³.

5.7.2 Laboratorio Remoto. La experimentación con una planta real es insustituible por una simulación o por simuladores de entrenamiento, sobre todo en cuanto a las sensaciones que percibe el usuario que se sitúa ante el experimento. Pero los laboratorios de prácticas físicas, tal y como se conciben actualmente, están sometidos a una serie de factores que limitan su aplicación ante determinadas situaciones, como son:

- **Elevado número de alumnos matriculados en la materia.** Esta restricción obliga al establecimiento de multitud de turnos y horarios, con perjuicio para los estudiantes: carencia de tiempo en los laboratorios para completar sus trabajos, ejecución de pocos casos prácticos, número elevado de personas por puesto de trabajo, etc.
- **Modelos educativos basados en la enseñanza a distancia.** Esta modalidad de estudio ocasiona que los alumnos lleguen a trasladarse durante días o semanas al lugar en el que se realizan las prácticas.
- **Carencia de recursos económicos.** En muchos casos condiciona que el número de experimentos disponibles en el laboratorio sea limitado.

Para resolver o aliviar en cierta medida los problemas planteados es necesario recurrir a un concepto ya estudiado desde hace algunos años, y que está accesible al estudiante y al público en general gracias a la difusión actual de la red *Internet*: la tele-operación. De forma genérica, el término tele-operación se puede

⁷³ Departamento de Informática y Automática., Op. Cit. Disponible en internet: <<http://lab.dia.uned.es/rlab/index.html>>

definir como la posibilidad de acceder a manipular y controlar a distancia ciertos recursos con las mismas posibilidades que se tendría si se operara sobre ellos de forma local, manual y directa.

En el contexto, se enfoca el uso del término tele-operación al acceso de los elementos de un laboratorio de prácticas utilizando los recursos que brinda la red *Internet* y con un nivel de presencia suficiente para poder desarrollar las actividades prácticas con la misma validez a como si estas se desarrollasen de la forma tradicional en las instalaciones del laboratorio. Si este término, la tele-operación, se aplica al presente caso, el acceso a los laboratorios presenciales sin restricciones espacio-temporales, surge el concepto del laboratorio remoto basado en el WWW⁷⁴.

⁷⁴ Ibíd., Disponible en internet: <<http://lab.dia.uned.es/rlab/index.html>>

6. DESARROLLO CONCEPTUAL

6.1 PROCESO DE IDENTIFICACIÓN DE NECESIDADES

6.1.1 Planteamiento de la misión

6.1.1.1 Descripción del producto. Plataforma que permite Tele-operar un manipulador robótico a través de *internet*.

6.1.1.2 Premisas y restricciones

- Tele-operar el manipulador robótico Scrobot-ER V plus, Scora-ER 14, Performer SV3X o Lab-Volt Armdroid MD 5200.
- Tele-operarlo a través de internet.
- Ubicación en el Laboratorio de Procesos de Manufactura.
- Protocolo de Comunicación RS232 entre el computador y el robot.

6.1.1.3 Partes implicadas. Universidad Autónoma de Occidente:

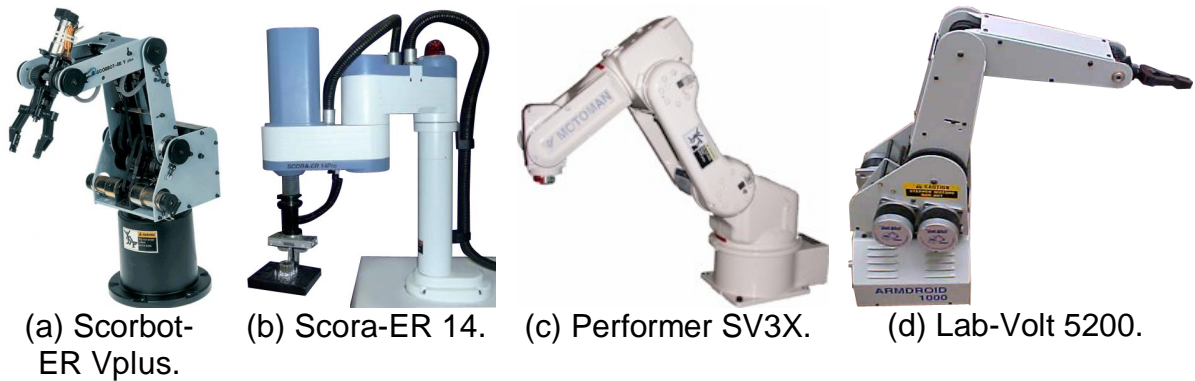
- Docentes.
- Estudiantes.

6.1.2 Obtención de datos primarios

6.1.2.1 Descripción de los Robots Manipuladores. La universidad cuenta con cuatro robots manipuladores, ubicados en el laboratorio de Procesos de

Manufactura, cada uno dotado de su respectivo computador, controlador y Teach Pendant. En la figura 10 se muestran todos los Robots Manipuladores disponibles.

Figura 10. Robots Manipuladores



Fuente: (a) TERA IMPEX. Scorbot ER5plus [en línea]. [Consultado el 02 de Septiembre del 2011]. Disponible en internet: <http://www.teraimpex.ro/detaliimaterial-didactic-universitar/f/scorbot-er5plus_246_6827.html> (b) SOLUCIONES TECNOLÓGICAS INTEGRADAS S.L. (STI). ROBÓTICA INDUSTRIAL [en línea]. [Consultado el 02 de Septiembre del 2011]. Disponible en internet: <http://www.stieducacion.com/index.php/STI/productos/P102/robotica_industrial> (c) CASARINI ROBÓTICA. *Industrial robot systems for welding and handling* [en línea]. [Consultado el 02 de Septiembre del 2011]. Disponible en internet: <http://casarinirobot.com/robot_sv3.html> (d) THE OLD ROBOTS WEB SITE. Lab Volt 5100 Robot [en línea]. [Consultado el 02 de Septiembre del 2011]. Disponible en internet: <<http://www.theoldrobots.com/Lab-Volt-5100.html>>

En su entorno de trabajo los robots presentan varios problemas, algunos inmersos en el diseño del robot, que no se pueden solucionar, y otros, relacionados con su ambiente laboral, que si se pueden atender. Estos problemas son:

- Los cuatro robots están ubicados, de tal forma que, hay objetos dentro de su espacio de trabajo, como lo son: su respectivo computador, otro robot o accesorio de este y “paredes” (modulo separador con vidrios).
- No hay Guardas de Seguridad o barreras físicas diseñadas como parte de, o agregadas a la máquina, que prevengan el acceso a las zonas peligrosas de esta, o señalización que delimiten el área de trabajo de los manipuladores haciendo potencialmente peligrosa la interacción con estos.

- El manipulador Lab-Volt 5200 presenta diversas colisiones con sus partes cuando se mueve a lo largo de su área de trabajo, impidiendo el movimiento continuo de sus elementos.

El manipulador Lab-Volt 5200 no tiene una información sobre sus características técnicas, tales como: rangos de movimiento, velocidades, carga útil, en otras palabras, sus limitaciones físicas. La empresa Lab-Volt tampoco dispone de ellas. Para calcularlas hay que operar el robot y descubrirlas. La figura 11 permite observar los problemas que hay en el ambiente de trabajo del robot.

Figura 11. Ambiente de trabajo del robot Performer SV3X



Las principales características de los cuatro Robots Manipuladores se muestran en el anexo E. Además, se promedian dichas características obteniendo una Referencia y más adelante se compara con los Robots con el fin de escoger el Manipulador con el cual se desea trabajar.

6.1.3 Interpretación de datos primarios. A continuación, se escogen las principales características del anexo E, de acuerdo al criterio que presenta Barrientos⁷⁵ en la Tabla 9.1 de su libro, para asignarles una nota entre uno y cinco en relación a la Referencia. Cada nota tiene un significado cualitativo, a saber en el cuadro 3:

⁷⁵ BARRIENTOS, Op. Cit., p. 262.

Cuadro 3. Cuadro de Ponderación

1	2	3	4	5
Mucho peor que.	Peor que.	Igual a.	Mejor que.	Mucho mejor que.

Una vez calificado se obtiene el Criterio Ponderado a partir del producto entre la Nota y el Porcentaje Ponderado, y así, sumar cada Criterio Ponderado para escoger el Robot de mayor Total. El Porcentaje Ponderado es una variable entre 0 – 100% que asigna la importancia de su respectiva característica frente a las otras siendo la suma total de Porcentajes Ponderados igual al 100%. Los resultados se pueden apreciar en el cuadro 4.

Para el efecto de no exceder 10 características en el cuadro 4 se hace un promedio de los rangos de movimiento de los tres primeros GDL de cada robot, siendo solamente los tres primeros, dado que, son los básicos para posicionar una pieza en el espacio.

Cuadro 4. Matriz para evaluar los Robots Manipuladores.

			SCORBOT-ER V PLUS		SCORA-ER 14		PERFORMER SV3X		LAB-VOLT 5200	
N° CRITERIOS	CRITERIO DE SELECCIÓN	PORCENTAJE PONDERACION	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO
1	Grados de Libertad	12%	3	0.360	2	0.240	4	0.480	3	0.360
2	Repetitividad de Posición	08%	3	0.240	4	0.320	5	0.400	1	0.080
3	Área de Trabajo Horizontal Aprox.	12%	2	0.240	1	0.120	4	0.480	2	0.240
4	Área de Trabajo Vertical Aprox.	12%	2	0.240	1	0.120	4	0.480	5	0.600
5	Apertura del Gripper	08%	3	0.240	2	0.160	2	0.160	4	0.320
6	Max Carga Útil	08%	1	0.080	2	0.160	4	0.320	5	0.400
7	Media Rango de Movimiento Eje 1 – 3	13%	4	0.520	2	0.260	5	0.650	3	0.390
8	Programación	13%	3	0.390	3	0.390	3	0.390	2	0.260
9	Comunicación	14%	3	0.420	3	0.420	3	0.420	3	0.420
Total			2.73		2.19		3.78		3.07	

			SCORBOT-ER V PLUS		SCORA-ER 14		PERFORMER SV3X		LAB-VOLT 5200	
N° CRITERIOS	CRITERIO DE SELECCIÓN	PORCENTAJE PONDERACION	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO
¿Continuar?			NO		NO		DESARROLLAR		NO	

Lo anterior se repite para el cuadro 5 en donde se califican las características de los lenguajes de programación. La cantidad de criterios expuestos en el anexo F se reducen a 10 escogiendo los más importantes. En el nivel de Caper Jones no se encontró el puntaje correspondiente a PHP y JavaScript, por ello, se les asignó dos. Para entender a que se refiere el Nivel Carpers Jones consulte el anexo G.

Cuadro 5. Matriz para evaluar los Lenguajes de Programación.

			JAVA		C		C#		C++		OBJECT IVE-C		PHP		VISUAL BASIC		PYTHON		PERL		JAVA SCRIPT	
N° CRITERIOS	CRITERIO DE SELECCIÓN	PORCENTAJE PONDERACION	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO	NOTA	CRITERIO PONDERADO
1	Popularidad 2012	10%	5	0.500	5	0.500	4	0.400	4	0.400	4	0.400	2	0.200	2	0.200	2	0.200	1	0.100	1	0.100
2	Estandarizado	15%	3	0.450	3	0.450	3	0.450	3	0.450	1	0.150	1	0.150	1	0.150	3	0.450	1	0.150	3	0.450
3	Orientado a Objetos	10%	3	0.300	1	0.100	3	0.300	3	0.300	3	0.300	3	0.300	2	0.200	3	0.300	3	0.300	1	0.100
4	Tipado Estático / Dinámico	5%	3	0.150	3	0.150	3	0.150	3	0.150	3	0.150	2	0.100	3	0.150	2	0.100	2	0.100	2	0.100
5	Integración de Lenguaje	10%	4	0.400	3	0.300	4	0.400	4	0.400	3	0.300	2	0.200	3	0.300	4	0.400	4	0.400	1	0.100
6	Múlti-hilo	5%	3	0.150	2	0.100	3	0.150	2	0.100	2	0.100	1	0.050	1	0.050	3	0.150	1	0.050	2	0.100
7	Seguridad Incorporada	10%	4	0.400	3	0.300	4	0.400	3	0.300	3	0.300	4	0.400	3	0.300	3	0.300	4	0.400	3	0.300
8	Nivel Capers Jones	10%	2	0.200	1	0.100	2	0.200	2	0.200	4	0.400	2	0.200	4	0.400	5	0.500	5	0.500	2	0.200
9	Licencia Libre	10%	3	0.300	3	0.300	3	0.300	3	0.300	3	0.300	3	0.300	1	0.100	4	0.400	3	0.300	3	0.300
10	Conocimiento del Programador	15%	3	0.450	1	0.150	1	0.150	1	0.150	1	0.150	1	0.150	2	0.300	1	0.150	1	0.150	1	0.150
TOTAL			3.30		2.45		2.90		2.75		2.55		2.00		2.10		2.95		2.40		1.90	
¿CONTINUAR?			SI		NO		NO		NO		NO		NO		NO		NO		NO		NO	

En el cuadro 6, se escribe lo que el cliente ha solicitado y lo que espera del producto tal como lo dijo, para posteriormente, traducirlo en necesidades. Para ello, hay que tener en cuenta que las necesidades dicen el qué y no el cómo, es decir, no dicen cómo se podría solucionar alguna necesidad; también, no se puede perder de vista que las necesidades se expresan como atributos del producto evitando usar frases negativas.

Cuadro 6. Planteamiento de las necesidades.

#	PLANTEAMIENTO DEL CLIENTE	PLANTEAMIENTO DE LA NECESIDAD
1	Los usuarios disponen de poco tiempo debido al trabajo, carga académica, situaciones personales, entre otras; y el hecho de desplazarse hasta la universidad implica tiempo adicional y dinero.	Puede operar al robot por mando a distancia.
2	La universidad tiene un horario de atención, el cual no se puede exceder, limitando la permanencia de la comunidad autónoma dentro de sus instalaciones.	Permite que el robot pueda estar disponible 24 horas al día.
3	Que el estudiante NO se lastime al usarlo.	Es seguro para el usuario.
4	Que tenga paro de emergencia.	Permite hacer Paro de Emergencia.
5	Que cumpla con las normas necesarias.	Cumple con las normas técnicas.
6	Que garantice que el robot sea operado dentro de las condiciones de trabajo para la cual fue diseñado.	Asegura la operación correcta del robot.
7	Que sea fácil de implementar.	Es fácil de implementar.
8	Que la solución sea económica.	Es económico.
9	El sistema debe funcionar con voltaje alterno a 110/220VAC.	Funciona con energía disponible.
10	Que sea fácil su mantenimiento.	De fácil mantenimiento.
11	Que la solución tenga una apariencia estética.	De apariencia estética.
12	Que pueda usarlo por mucho tiempo.	Vida útil extensa.
13	Que sea compatible diferentes Sistemas Operativos	Es multiplataforma.
14	Que el robot se pueda operar en tiempo real.	La interacción es en tiempo real.
15	Que se pueda ver cuando el robot es operado.	Se puede visualizar el robot.
16	Se puedan enviar rutinas al robot.	Se pueden realizar rutinas.
17	Que el robot NO se dañe al manipularlo.	Es seguro para el robot.

Las anteriores necesidades poseen una relación de importancia entre ellas, para saberlo, se organiza un cuadro en donde se familiaricen y califiquen una frente a la otra de acuerdo al cuadro 7.

Cuadro 7. Cuadro de Ponderación para Matriz de Comparación Pareada.

CALIFICACIÓN	DESCRIPCIÓN
1	El elemento Columna y Fila tienen la misma importancia.
3	El elemento Columna es ligeramente más importante que el elemento Fila.
5	El elemento Columna es más importante que el elemento Fila.
7	El elemento Columna es significativamente más importante que el elemento Fila.
9	El elemento Columna es extremadamente más importante que el elemento Fila.
1/3	El elemento Fila es ligeramente más importante que el elemento Columna.
1/5	El elemento Fila es más importante que el elemento Columna.
1/7	El elemento Fila es significativamente más importante que el elemento Columna.
1/9	El elemento Fila es extremadamente más importante que el elemento Columna.

Y así, obtener el cuadro 8 conocido como “Matriz de Comparación Pareada” que contiene una columna “SUMATORIA RENGLÓN”, la cual, es el resultado de sumar todos los elementos de una fila. Observe el cálculo del primer elemento de esta columna en la ecuación 1.

Ecuación 1. Cálculo del primer elemento de la Columna Sumatoria Renglón

$$\sum_{i=1}^{17} F_i = 1 + \frac{1}{5} + 7 + \frac{1}{3} + 7 + \frac{1}{5} + \frac{1}{7} + \frac{1}{3} + \frac{1}{3} + \frac{1}{5} + \frac{1}{9} + \frac{1}{3} + \frac{1}{7} + \frac{1}{3} + 1 + \frac{1}{9} + 7 = 25,77$$

Cuadro 8. Matriz de Comparación Pareada

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	SUMATORIA RENGLÓN
		NECESIDADES	Puede operar al robot por mando a distancia.	Permite que el robot pueda estar disponible 24 horas al día.	Es seguro para el usuario.	Permite hacer Paro de Emergencia.	Cumple con las normas técnicas.	Asegura la operación correcta del robot.	Es fácil de implementar.	Es económico.	Funciona con energía disponible.	De fácil mantenimiento.	De apariencia estética.	Vida útil extensa.	Es multiplataforma.	La interacción es en tiempo real.	Se puede visualizar el robot.	Se pueden realizar rutinas.	Es seguro para el robot.	
#	NECESIDADES																			
1	Puede operar al robot por mando a distancia.	1	1/5	7	1/3	7	1/5	1/7	1/3	1/3	1/5	1/9	1/3	1/7	1/3	1	1/9	7	25,77	
2	Permite que el robot pueda estar disponible 24 horas al día.	5	1	9	3	9	9	1/5	1/3	1/3	1/3	1/7	1/3	1/5	1/3	3	9	9	59,21	
3	Es seguro para el usuario.	1/7	1/9	1	1/5	1	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/7	1/7	1/9	1/9	3,85	
4	Permite hacer Paro de Emergencia.	3	1/3	5	1	3	1/3	1/7	1/5	1/5	1/5	1/7	1/5	1/7	1	1/3	1/7	1/3	15,70	
5	Cumple con las normas técnicas.	1/7	1/9	1	1/3	1	1/3	1/7	1/7	1/7	1/7	1/9	1/7	1/9	1/7	1/7	1/9	1/7	4,40	
6	Asegura la operación correcta del robot.	5	1/9	9	3	3	1	1/5	1/7	1/5	1/7	1/9	1/5	1/7	1/5	1/3	1/9	3	25,90	
7	Es fácil de implementar.	7	5	9	7	7	5	1	3	3	1	1/5	5	3	5	5	1/5	7	73,40	
8	Es económico.	3	3	9	5	7	7	1/3	1	1/5	1/5	1/7	1/3	1/5	3	3	1/9	7	49,52	
9	Funciona con energía disponible.	3	3	9	5	7	5	1/3	5	1	5	1/5	5	1/3	5	3	1/7	7	64,01	

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	SUMATORIA RENGLÓN
		NECESIDADES	Puede operar al robot por mando a distancia.	Permite que el robot pueda estar disponible 24 horas al día.	Es seguro para el usuario.	Permite hacer Paro de Emergencia.	Cumple con las normas técnicas.	Asegura la operación correcta del robot.	Es fácil de implementar.	Es económico.	Funciona con energía disponible.	De fácil mantenimiento.	De apariencia estética.	Vida útil extensa.	Es multiplataforma.	La interacción es en tiempo real.	Se puede visualizar el robot.	Se pueden realizar rutinas.	Es seguro para el robot.	
#	NECESIDADES																			
10	De fácil mantenimiento.	5	3	9	5	7	7	1	5	1/5	1	1/5	3	1/3	5	5	1/5	7	63,93	
11	De apariencia estética.	9	7	9	7	9	9	5	7	5	5	1	9	5	7	7	1/3	9	110,33	
12	Vida útil extensa.	3	3	9	5	7	5	1/5	3	1/5	1/3	1/9	1	1/5	5	3	1/7	1	46,19	
13	Es multiplataforma.	7	5	9	7	9	7	1/3	5	3	3	1/5	5	1	5	5	3	9	83,53	
14	La interacción es en tiempo real.	3	3	7	1	7	5	1/5	1/3	1/5	1/5	1/7	1/5	1/5	1	1/3	1/7	7	35,95	
15	Se puede visualizar el robot.	1	1/3	7	3	7	3	1/5	1/3	1/3	1/5	1/7	1/3	1/5	3	1	1/5	7	34,28	
16	Se pueden realizar rutinas.	9	1/9	9	7	9	9	5	9	7	5	3	7	1/3	7	5	1	9	101,44	
17	Es seguro para el robot.	1/7	1/9	9	3	7	1/3	1/7	1/7	1/7	1/7	1/9	1	1/9	1/7	1/7	1/9	1	22,78	

A continuación se procede a realizar la “Matriz de Normalización” en el cuadro 9, en ella se ubica el resultado de dividir cada elemento de la matriz anterior, por el valor correspondiente a su fila de la columna “SUMATORIA RENGLÓN”, por ejemplo, calculemos el primer elemento, (Fila uno, Columna uno), mostrado en la ecuación 2:

Ecuación 2. Cálculo del elemento Fila uno, Columna uno

$$\frac{F_1 C_1}{\sum_{i=1}^{17} F_i} = \frac{1}{25.77} = 0.0338$$

Cuadro 9. Matriz de Normalización.

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
		NECESIDADES	Puede operar al robot por mando a distancia.	Permite que el robot pueda estar disponible 24 horas al día.	Es seguro para el usuario.	Permite hacer Paro de Emergencia.	Cumple con las normas técnicas.	Asegura la operación correcta del robot.	Es fácil de implementar.	Es económico.	Funciona con energía disponible.	De fácil mantenimiento.	De apariencia estética.	Vida útil extensa.	Es multiplataforma.	La interacción es en tiempo real.	Se puede visualizar el robot.	Se pueden realizar rutinas.	Es seguro para el robot.	
#	NECESIDADES																			
1	Puede operar al robot por mando a distancia.	0,0388	0,0078	0,2716	0,0129	0,2716	0,0078	0,0055	0,0129	0,0129	0,0078	0,0043	0,0129	0,0055	0,0129	0,0388	0,0043	0,2716	25,77	SUMATORIA RENGLÓN

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	SUMATORIA RENGLÓN
		NECESIDADES	Puede operar al robot por mando a distancia.	Permite que el robot pueda estar disponible 24 horas al día.	Es seguro para el usuario.	Permite hacer Paro de Emergencia.	Cumple con las normas técnicas.	Asegura la operación correcta del robot.	Es fácil de implementar.	Es económico.	Funciona con energía disponible.	De fácil mantenimiento.	De apariencia estética.	Vida útil extensa.	Es multiplataforma.	La interacción es en tiempo real.	Se puede visualizar el robot.	Se pueden realizar rutinas.	Es seguro para el robot.	
#	NECESIDADES	0,0844	0,0169	0,1520	0,0507	0,1520	0,1520	0,0034	0,0056	0,0056	0,0056	0,0024	0,0056	0,0034	0,0056	0,0507	0,1520	0,1520	59,21	
2	Permite que el robot pueda estar disponible 24 horas al día.	0,0371	0,0289	0,2597	0,0519	0,2597	0,0289	0,0289	0,0289	0,0289	0,0289	0,0289	0,0289	0,0289	0,0371	0,0371	0,0289	0,0289	3,85	
3	Es seguro para el usuario.	0,1910	0,0212	0,3184	0,0637	0,1910	0,0212	0,0091	0,0127	0,0127	0,0127	0,0091	0,0127	0,0091	0,0637	0,0212	0,0091	0,0212	15,70	
4	Permite hacer Paro de Emergencia.	0,0325	0,0253	0,2274	0,0758	0,2274	0,0758	0,0325	0,0325	0,0325	0,0325	0,0253	0,0325	0,0253	0,0325	0,0325	0,0253	0,0325	4,40	
5	Cumple con las normas técnicas.	0,1931	0,0043	0,3476	0,1159	0,1159	0,0386	0,0077	0,0055	0,0077	0,0055	0,0043	0,0077	0,0055	0,0077	0,0129	0,0043	0,1159	25,90	
6	Asegura la operación correcta del robot.	0,0954	0,0681	0,1226	0,0954	0,0954	0,0681	0,0136	0,0409	0,0409	0,0136	0,0027	0,0681	0,0409	0,0681	0,0681	0,0027	0,0954	73,40	
7	Es fácil de implementar.	0,0606	0,0606	0,1817	0,1010	0,1414	0,1414	0,0067	0,0202	0,0040	0,0040	0,0029	0,0067	0,0040	0,0606	0,0606	0,0022	0,1414	49,52	
8	Es económico.	0,0469	0,0469	0,1406	0,0781	0,1094	0,0781	0,0052	0,0781	0,0156	0,0781	0,0031	0,0781	0,0052	0,0781	0,0469	0,0022	0,1094	64,01	
9	Funciona con energía disponible.	0,0782	0,0469	0,1408	0,0782	0,1095	0,1095	0,0156	0,0782	0,0031	0,0156	0,0031	0,0469	0,0052	0,0782	0,0782	0,0031	0,1095	63,93	
10	De fácil mantenimiento.	0,0816	0,0634	0,0816	0,0634	0,0816	0,0816	0,0453	0,0634	0,0453	0,0453	0,0091	0,0816	0,0453	0,0634	0,0634	0,0030	0,0816	110,33	
11	De apariencia estética.																			

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
		NECESIDADES	Puede operar al robot por mando a distancia.	Permite que el robot pueda estar disponible 24 horas al día.	Es seguro para el usuario.	Permite hacer Paro de Emergencia.	Cumple con las normas técnicas.	Asegura la operación correcta del robot.	Es fácil de implementar.	Es económico.	Funciona con energía disponible.	De fácil mantenimiento.	De apariencia estética.	Vida útil extensa.	Es multiplataforma.	La interacción es en tiempo real.	Se puede visualizar el robot.	Se pueden realizar rutinas.	Es seguro para el robot.	SUMATORIA RENGLÓN
#	NECESIDADES																			
12	Vida útil extensa.		0,0650	0,0650	0,1949	0,1083	0,1516	0,1083	0,0043	0,0650	0,0043	0,0072	0,0024	0,0217	0,0043	0,1083	0,0650	0,0031	0,0217	46,19
13	Es multiplataforma.		0,0838	0,0599	0,1077	0,0838	0,1077	0,0838	0,0040	0,0599	0,0359	0,0359	0,0024	0,0599	0,0120	0,0599	0,0599	0,0359	0,1077	83,53
14	La interacción es en tiempo real.		0,0834	0,0834	0,1947	0,0278	0,1947	0,1391	0,0056	0,0093	0,0056	0,0056	0,0040	0,0056	0,0056	0,0278	0,0093	0,0040	0,1947	35,95
15	Se puede visualizar el robot.		0,0292	0,0097	0,2042	0,0875	0,2042	0,0875	0,0058	0,0097	0,0097	0,0058	0,0042	0,0097	0,0058	0,0875	0,0292	0,0058	0,2042	34,28
16	Se pueden realizar rutinas.		0,0887	0,0011	0,0887	0,0690	0,0887	0,0887	0,0493	0,0887	0,0690	0,0493	0,0296	0,0690	0,0033	0,0690	0,0493	0,0099	0,0887	101,44
17	Es seguro para el robot.		0,0063	0,0049	0,3951	0,1317	0,3073	0,0146	0,0063	0,0063	0,0063	0,0063	0,0049	0,0439	0,0049	0,0063	0,0063	0,0049	0,0439	22,78
PROMEDIO COLUMNA			0,0762	0,0361	0,2017	0,0762	0,1652	0,0779	0,0146	0,0363	0,0200	0,0003	0,0084	0,0348	0,0126	0,0510	0,0429	0,0177	0,1071	0,98
PROMEDIO PORCENTAJE			7,62	3,61	20,17	7,62	16,52	7,79	1,46	3,63	2,00	0,03	0,84	3,48	1,26	5,10	4,29	1,77	10,71	97,91
PRIORIDAD			5	10	1	6	2	4	14	9	12	17	16	11	15	7	8	13	3	

Finalizando la matriz se encuentran tres filas, la primera es “PROMEDIO COLUMNA”, cuyo valor es la media de los datos de su respectiva columna (ecuación 3).

Ecuación 3. Cálculo de la Media de los datos de la primera Columna

$$\frac{\sum_{i=1}^{17} C_i}{17} = \frac{1,296}{17} = 0.0762$$

Donde

$$\sum_{i=1}^{17} C_i = 0.0388 + 0.0844 + 0.0371 + 0.1910 + 0.0325 + 0.1931 + 0.0954 + 0.0606 + 0.0469 + 0.0782 + 0.0816 + 0.0650 + 0.0838 + 0.0834 + 0.0292 + 0.0887 + 0.0063 = 1,296$$

La segunda es “PROMEDIO PORCENTAJE”, que es el porcentaje “PROMEDIO COLUMNA”.

Y la tercera es “PRIORIDAD”, que es “PROMEDIO PORCENTAJE” en orden descendente, dando el valor de uno (1) al mayor y 17 al menor, como se puede visualizar en el cuadro 10.

Cuadro 10. Prioridad de las necesidades.

PRIORIDAD	NECESIDADES	PROMEDIO %
1	Es seguro para el usuario.	20,17
2	Cumple con las normas técnicas.	16,52
3	Es seguro para el robot.	10,71
4	Asegura la operación correcta del robot.	7,79
5	Puede operar al robot por mando a distancia.	7,62
6	Permite hacer Paro de Emergencia.	7,62
7	La interacción es en tiempo real.	5,10
8	Se puede visualizar el robot.	4,29
9	Es económico.	3,63
10	Permite que el robot pueda estar disponible 24 horas al día.	3,61
11	Vida útil extensa.	3,48
12	Funciona con energía disponible.	2,00
13	Se pueden realizar rutinas.	1,77
14	Es fácil de implementar.	1,46
15	Es multiplataforma.	1,26
16	De apariencia estética.	0,84
17	De fácil mantenimiento.	0,03

6.2 ESTABLECER ESPECIFICACIONES DEL PRODUCTO

6.2.1 Establecer métricas y sus unidades. El fin de esta sección es determinar las especificaciones técnicas preliminares, en otras palabras, se bosquejan las variables que definen el producto; todas ellas son el producto de extraer las variables implícitas en las necesidades del cliente usando el sentido común. El cuadro 11 relaciona las métricas, las unidades y su correspondencia con las necesidades del cliente.

Cuadro 11. Establecer métricas y sus unidades

N°	Métrica	Unidades	N° Necesidades
1	Accesibilidad Remota	Lugar	1, 5, 7, 9, 10, 13
2	Tiempo Disponible por día	Hora [h]	5, 9, 10, 11
3	Compatibilidad	Lista (Sistema Operativo, Hardware)	4, 5, 7, 9, 11, 12, 13, 14, 15, 17
4	Velocidad de Transmisión	Kilobytes / Segundo [Kb/s]	5, 7, 8, 13
5	Fuerza Electromotriz	Voltio [V] (Depende del país)	1, 2, 3, 12, 17
6	Frecuencia	Hercio [Hz] (Depende del país)	1, 2, 3, 12, 17
7	Corriente	Amperio [A]	1, 2, 3, 12, 17
8	Consumo de Energía	Kilowatio-hora [Kw-h]	9, 10
9	Comunicación	Lista de Protocolos	2, 3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17
10	Programación	Lista de Programas	2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17
11	Nivel de Ruido	Decibel [dB]	1, 2, 10
12	Vida Útil	Años, Meses	9, 10, 11
13	Seguro	Subjetivo	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14, 17
14	Normas	Lista de Normas	1, 2, 3, 4, 6, 9, 10, 11, 12, 14, 17
15	Fácil Mantenimiento	Subjetivo	9, 10, 12, 14, 17
16	Estética	Subjetivo	9, 16
17	Costo de Uso	Peso [\$], Dólar US	3, 9, 10, 12, 14
18	Costo de Adquisición	Peso [\$], Dólar US	9, 12, , 14, 16
19	Costo de Mantenimiento	Peso [\$], Dólar US	3, 9, 10, 12, 14, 17
20	Costo de Reparación	Peso [\$], Dólar US	3, 6, 9, 10, 12, 14, 17

En el anexo H se presenta un cuadro que relaciona países, voltajes y frecuencias, correspondientes a las métricas “Accesibilidad Remota”, “Fuerza Electromotriz” y “Frecuencia”, respectivamente. En el anexo I se muestran las listas correspondientes al valor de las unidades de “Compatibilidad”, “Comunicación”, “Programación” y “Normas”.

6.2.2 Relacionar las métricas entre sí. En el cuadro 12 se relacionan las métricas dos a dos, asignando el símbolo más (+) cuando la mejora o incremento de una métrica afecta positivamente a la otra, el signo menos (-) cuando la mejora de una métrica compromete a la otra negativamente y cero (0) cuando se relaciona una métrica consigo misma o si no hay efecto alguno de una sobre la otra (para efecto de facilitar la lectura, se deja en blanco cuando se asigna el cero). A este cuadro se le adiciona una columna en donde se restan la cantidad de símbolos más (+) con la cantidad de símbolos menos (-), esto permite tener en cuenta hasta donde se puede enfocar la mejora de una métrica sin comprometer otra. Entre más positivo sea el resultado y más arriba se encuentre en el cuadro, más fácil se puede mejorar una métrica; y entre más negativa y más abajo se encuentre en el cuadro, más fácil se puede disminuir la métrica. En el caso de que se pueda y se decida mejorar una métrica se le asocia el símbolo (↑), si se altera negativamente le corresponde el símbolo (↓), y si se mantiene, se establece (●).

Cuadro 12. Matriz de correlación de las métricas.

#	MÉTRICAS	Simb	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	SUMATORIA RENGLÓN
			MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación	
1	Accesibilidad Remota	↑			+	+	+				-	+	+		+	+				-	-	-		3
2	Tiempo Disponible por día	↑	+			+	+					+	+	-	-	-				-		-	-	-1
3	Compatibilidad	↑	+	+			+	+	+	+	+	+	+		+	+	+	+		+	+	+	+	17
4	Velocidad de Transmisión	●	+	+	+							+	+		+		-							5
5	Fuerza Electromotriz	●			+					-	-	+			+	-	-	-		-	-	-	-	-6

#	MÉTRICAS	Simb	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	SUMATORIA RENGLÓN
			MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación	
6	Frecuencia	•			+										+	+	+	+						5
7	Corriente	•			+			-			-				-	-	-	-		-	-	-	-	-9
8	Consumo de Energía	•		-	-	+		-		-		-					-			-				-6
9	Comunicación	↑		+	+	+	+	+			-		+		+		+	+				+	+	10
10	Programación	↑		+	+	+	+					+			+	+	+	+		-	-	+	+	9
11	Nivel de Ruido	•			-											+	+							1
12	Vida Útil	•		+	-	+	+	+	+	-		+	+			+	+				-			6
13	Seguro	•		+	-	+		-	+	-			+	+	+		+	+		+	-			6
14	Normas	•				+	-	-	+	-	-	+	+	+	+	+		+			-	+	+	4
15	Fácil Mantenimiento	•				+		-	+	-		+	+			+	+					+	+	6
16	Estética	•																			-			-1
17	Costo de Uso	↓		-	-	+		-		-	-		-			+						-	-	-6
18	Costo de Adquisición	↓		-		+		-		-			-		-	-	-		-					-7
19	Costo de Mantenimiento	↓		-	-	+		-		-		+	+				+	+		-				0
20	Costo de Reparación	↓			-	+		-		-		+	+				+	+		-				1

6.2.3 Relacionar las métricas con las necesidades. Permite observar el aporte que brinda cada métrica para el cumplimiento de alguna necesidad del cliente, permitiendo identificar puntos débiles y fuertes. En el cuadro 13 se relacionan las métricas con las necesidades calificando esta relación con nueve (9) cuando es fuerte, tres (3) para media, uno (1) si es débil y cero (0) cuando es nula (para efecto de facilitar la lectura, se deja en blanco cuando se asigne el cero).

Cuadro 13. Relacionar las métricas con las necesidades

#	NECESIDADES	Imp	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	SUMATORIA RENGLÓN
			MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación	
1	Es seguro para el usuario.	20,17	9					3	3	9				3		9	9							9,077
2	Cumple con las normas técnicas.	16,52						9	9	9		3	3	9		9	9							9,912
3	Es seguro para el robot.	10,71						9	9	9		3	3			9	9			3		9	9	7,711
4	Asegura la operación correcta del robot.	7,79				3		3	3	3			9			9	9							3,038
5	Puede operar al robot por mando a distancia.	7,62	9	9	3	3						9	9											3,200
6	Permite hacer Paro de Emergencia.	7,62														9	9						1	1,448
7	La interacción es en tiempo real.	5,10	3			3	9					9	3			9								1,836

#	NECESIDADES	Imp	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	SUMATORIA RENGLÓN
			MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación	
8	Se puede visualizar el robot.	4,29				9						9	1			3								0,944
9	Es económico.	3,63	3	9	3						9	1	9		3	1	3	3	3	9	9	9	9	3,013
10	Permite que el robot pueda estar disponible 24 horas al día.	3,61	9	9							9	3	3	1	3	9	9	3		3		3	3	2,419
11	Vida útil extensa.	3,48		9	3							3	3		9	9	9							1,566
12	Funciona con energía disponible.	2,00			9		9	9	9			9	9				9	3		3	3	3	3	1,560
13	Se pueden realizar rutinas.	1,77	3		9	9						9	9											0,690
14	Es fácil de implementar.	1,46			3							9	9			9	9	9		3	9	3	3	0,964
15	Es multiplataforma.	1,26			9							9	9											0,340
16	De apariencia estética.	0,84																	9		3			0,101
17	De fácil mantenimiento.	0,03			9		3	3	3			9	9			9	9	9				9	9	0,024

6.2.4 Importancia técnica de las métricas. Informa la importancia de una métrica para satisfacer las necesidades del cliente, permitiendo jerarquizarlas. Se calcula multiplicando la importancia de cada necesidad por el valor numérico (9, 3, 1, o 0) asignado y se suma el resultado de todas las multiplicaciones realizadas sobre esa columna (para efecto de facilitar la lectura, se deja en blanco cuando el resultado sea cero). La ponderación total correspondiente a cada métrica permite saber cuáles deben ser consideradas con prioridad.

En adición, se incorpora otras dos filas; en la primera, “PORCENTAJE”, se imprime el valor relativo de las ponderaciones de las métricas (importancia técnica relativa) que se calcula dividiendo la ponderación correspondiente a cada métrica entre el resultado de la suma de todas las ponderaciones multiplicado por 100; y, en la otra, “PRIORIDAD”, se organiza en orden descendente “PORCENTAJE” dándole un valor de uno (1) a la métrica de mayor resultado y de 20 a la de menor resultado, jerarquizando de esta manera las métricas. Lo anterior se puede analizar en el cuadro 14.

Cuadro 14. Importancia Técnica de las métricas

#	NECESIDADES	Imp	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación
1	Es seguro para el usuario.	20,17		1,815				0,605	0,605	1,815				0,605		1,815	1,815						

#	NECESIDADES	Imp	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación
2	Cumple con las normas técnicas.	16,52						1,487	1,487	1,487		0,496	0,496	1,487		1,487	1,487						
3	Es seguro para el robot.	10,71						0,964	0,964	0,964		0,321	0,321			0,964	0,964			0,321		0,964	0,964
4	Asegura la operación correcta del robot.	7,79				0,234		0,234	0,234	0,234			0,701			0,701	0,701						
5	Puede operar al robot por mando a distancia.	7,62	0,686	0,686	0,229	0,229						0,686	0,686										
6	Permite hacer Paro de Emergencia.	7,62														0,686	0,686						0,076
7	La interacción es en tiempo real.	5,10	0,153			0,153	0,459					0,459	0,153			0,459							
8	Se puede visualizar el robot.	4,29					0,386					0,386	0,043			0,129							
9	Es económico.	3,63	0,109	0,327	0,109						0,327	0,036	0,327		0,109	0,036	0,109	0,109	0,109	0,327	0,327	0,327	0,327
10	Permite que el robot pueda estar disponible 24 horas al día.	3,61	0,325	0,325							0,325	0,108	0,108	0,036	0,108	0,325	0,325	0,108		0,108		0,108	0,108

		#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
		MÉTRICAS	Accesibilidad Remota	Tiempo Disponible por día	Compatibilidad	Velocidad de Transmisión	Fuerza Electromotriz	Frecuencia	Corriente	Consumo de Energía	Comunicación	Programación	Nivel de Ruido	Vida Útil	Seguro	Normas	Fácil Mantenimiento	Estética	Costo de Uso	Costo de Adquisición	Costo de Mantenimiento	Costo de Reparación
#	NECESIDADES	Imp																				
11	Vida útil extensa.	3,48		0,313	0,104						0,104	0,104		0,313	0,313	0,313						
12	Funciona con energía disponible.	2,00			0,180		0,180	0,180	0,180		0,180	0,180				0,180	0,060		0,060	0,060	0,060	0,060
13	Se pueden realizar rutinas.	1,77	0,053		0,159	0,159					0,159	0,159										
14	Es fácil de implementar.	1,46			0,044						0,131	0,131			0,131	0,131	0,131		0,044	0,131	0,044	0,044
15	Es multiplataforma.	1,26			0,113						0,113	0,113										
16	De apariencia estética.	0,84																0,076		0,025		
17	De fácil mantenimiento.	0,03			0,003		0,001	0,001	0,001		0,003	0,003			0,003	0,003	0,003				0,003	0,003
SUMATORIA COLUMNA			3,141	1,651	1,328	1,233	3,470	3,470	4,681	0,652	3,184	3,526	2,128	0,530	7,049	6,714	0,411	0,185	0,860	0,543	1,505	1,582
PORCENTAJE			6,565	3,450	2,775	2,577	7,254	7,254	9,783	1,362	6,654	7,370	4,448	1,109	14,734	14,034	0,860	0,386	1,798	1,136	3,147	3,306
PRIORIDAD			8	10	13	14	5	6	3	16	7	4	9	18	1	2	19	20	15	17	12	11

* La prioridad va de menor a mayor, siendo uno (1) el de mayor importancia hasta 20, cuya relevancia es menor.

6.2.5 Evaluar satisfacción del cliente en productos competidores. Es importante tener en cuenta los productos que ya existen en el mercado y cuales ha usado el cliente, para tomar como referencia el producto con mayor satisfacción y los productos con mejores características, con el fin de desarrollar un producto a la medida del cliente. Observar cuadro 15.

Cuadro 15. Evaluar satisfacción del cliente en productos competidores

#	NECESIDADES	IMPORTANCIA	Hyperterminal	TeraTerm	ATS (Advanced Terminal Software)
1	Es seguro para el usuario.	20,17	3	5	3
2	Cumple con las normas técnicas.	16,52	4	5	4
3	Es seguro para el robot.	10,71	4	4	5
4	Asegura la operación correcta del robot.	7,79	4	4	5
5	Puede operar al robot por mando a distancia.	7,62	1	5	1
6	Tiene Paro de Emergencia.	7,62	1	1	1
7	La interacción es en tiempo real.	5,10	5	5	5
8	Se puede visualizar el robot.	4,29	1	1	1
9	Es económico.	3,63	5	5	5
10	Permite que el robot pueda estar disponible 24 horas al día.	3,61	3	4	4
11	Vida útil extensa.	3,48	5	4	3
12	Funciona con energía disponible.	2,00	5	5	5
13	Se pueden realizar rutinas.	1,77	5	5	5
14	Es fácil de implementar.	1,46	5	5	4
15	Es multiplataforma.	1,26	1	5	1
16	De apariencia estética.	0,84	5	4	3
17	De fácil mantenimiento.	0,03	5	4	3

6.2.6 Evaluar métricas en productos competidores. Conocer las métricas y sus valores en los productos del mercado aterriza los posibles valores que se quisieran obtener de las especificaciones técnicas preliminares del producto a desarrollar para ser competitivo. Ver cuadro 16.

Cuadro 16. Evaluar métricas en productos competidores

N° Métricas	N° Necesidades	Métrica	Unidades	Importancia	Hyperterminal	TeraTerm	ATS (Advanced Terminal Software)
1	1, 5, 7, 9, 10, 13	Accesibilidad Remota	Lugar	6,565	Local	Local/ Remoto	Local
2	5, 9, 10, 11	Tiempo Disponible por día	Hora [h]	3,450	13,5	24	13,5
3	4, 5, 7, 9, 11, 12, 13, 14, 15, 17	Compatibilidad	Lista (Sistema Operativo, Hardware)	2,775	Windows 95 hasta 7	Windows 95 hasta 7	Windows 95 hasta 7
4	5, 7, 8, 13	Velocidad de Transmisión	Kilobytes / Segundo [Kb/s]	2,577	0	> 12800	0
5	1, 2, 3, 12, 17	Fuerza Electromotriz	Voltio [V] (Depende del país)	7,254	100 - 250	100 - 250	100 – 250
6	1, 2, 3, 12, 17	Frecuencia	Hercio [Hz] (Depende del país)	7,254	50/60	50/60	50/60
7	1, 2, 3, 12, 17	Corriente	Amperio [A]	9,783	x	x	x
8	9, 10	Consumo de Energía ⁷⁶	Kilowatio-hora [Kw-h]	1,362	0.2 Kw-h	0.2 Kw-h	0.2 Kw-h
9	2, 3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17	Comunicación	Lista de Protocolos	6,654	RS232	RS232	RS232
10	2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17	Programación	Lista de Programas	7,370	C	C	Visual Basic

⁷⁶ Ministerio de Minas y Energía. El Uso Racional y Eficiente de la Energía es responsabilidad de todos [en línea]. Bogotá, Colombia: Ministerio de Minas y Energía. Actualizado el 22 de Diciembre del 2009 [Consultado el 22 de marzo del 2012]. Disponible en internet: <http://www.minminas.gov.co/minminas/index.jsp?cargaHome=2&id_comunicado=616&opcionCalendar=10>

N° Métricas	N° Necesidades	Métrica	Unidades	Importancia	Hyperterminal	TeraTerm	ATS (Advanced Terminal Software)
11	1, 2, 10	Nivel de Ruido	Decibel [dB]	4,448	0 dB	0 dB	0 dB
12	9, 10, 11	Vida Útil	Años, Meses	1,109	7 años	10 años	5 años
13	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14, 17	Seguro	Subjetivo	14,734	Si	Si	Si
14	1, 2, 3, 4, 6, 9, 10, 11, 12, 14, 17	Normas	Lista de Normas	14,034	Cumple	Cumple	Cumple
15	9, 10, 12, 14, 17	Fácil Mantenimiento	Subjetivo	0,860	Si	Si	Si
16	1, 5, 7, 9, 10, 13	Estética	Subjetivo	0,386	Si	Si	Si
17	5, 9, 10, 11	Costo de Uso	Peso [\$], Dólar US	1,798	\$ 0	\$ 0	\$ 0
18	4, 5, 7, 9, 11, 12, 13, 14, 15, 17	Costo de Adquisición	Peso [\$], Dólar US	1,136	\$ 0	\$ 0	\$ 0
19	5, 7, 8, 13	Costo de Mantenimiento	Peso [\$], Dólar US	3,147	\$ 0	\$ 0	\$ 0
20	1, 2, 3, 12, 17	Costo de Reparación	Peso [\$], Dólar US	3,306	\$ 0	\$ 0	\$ 0

6.2.7 Asignar valores ideales y marginales. Aquí se establecen las especificaciones preliminares, es decir, se hacen las posibles especificaciones del producto para tener una idea de lo que se necesita obtener y al final, después de encontrar la solución al proyecto y terminarla, se procede a extraer las especificaciones finales. En el cuadro 17 se plasma lo anterior, en donde, los valores ideales son aquellos que hacen que la métrica sea optima y, los valores marginales son aquellos que se pueden aceptar, por ello, generalmente se da un rango admisible.

Cuadro 17. Especificaciones Preliminares

N° Métricas	N° Necesidades	Métrica	Unidades	Importancia	Valor Ideal	Valor Marginal
1	1, 5, 7, 9, 10, 13	Accesibilidad Remota	Lugar	6,565	Planeta Tierra	Cali - Colombia
2	5, 9, 10, 11	Tiempo Disponible por día	Hora [h]	3,450	24	15 - 24
3	4, 5, 7, 9, 11, 12, 13, 14, 15, 17	Compatibilidad	Lista	2,775	Todos	Windows Xp y 7
4	5, 7, 8, 13	Velocidad de Transmisión	Kilobytes / Segundo [Kbps]	2,577	> 12800	> 12800
5	1, 2, 3, 12, 17	Fuerza Electromotriz	Voltio [V]	7,254	100 - 250	110 y 220
6	1, 2, 3, 12, 17	Frecuencia	Hercio [Hz]	7,254	50 y 60	60
7	1, 2, 3, 12, 17	Corriente	Amperio [A]	9,783	?	?
8	9, 10	Consumo de Energía	Kilowatio-hora [Kw-h]	1,362	0.2	0.2 – 0.3
9	2, 3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17	Comunicación	Lista de Protocolos	6,654	RS232, TCP/IP y RTP	RS232 y TCP/IP
10	2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17	Programación	Lista de Programas	7,370	Java	Ver anexo I
11	1, 2, 10	Nivel de Ruido	Decibel [dB]	4,448	0	0
12	9, 10, 11	Vida Útil	Años, Meses	1,109	10 años	5 – 10 años
13	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14, 17	Seguro	Subjetivo	14,734	Si	Si
14	1, 2, 3, 4, 6, 9, 10, 11, 12, 14, 17	Normas	Lista de Normas	14,034	Cumple	Cumple
15	9, 10, 12, 14, 17	Fácil Mantenimiento	Subjetivo	0,860	Si	Si
16	1, 5, 7, 9, 10, 13	Estética	Subjetivo	0,386	Si	Si
17	5, 9, 10, 11	Costo de Uso	Peso [\$], Dólar US	1,798	0	0
18	4, 5, 7, 9, 11, 12, 13, 14, 15, 17	Costo de Adquisición	Peso [\$], Dólar US	1,136	0	0
19	5, 7, 8, 13	Costo de Mantenimiento	Peso [\$], Dólar US	3,147	0	0
20	1, 2, 3, 12, 17	Costo de Reparación	Peso [\$], Dólar US	3,306	0	0

6.3 GENERACIÓN DE CONCEPTOS

6.3.1 Clarificar el problema

6.3.1.1 Descripción del producto. Plataforma que permite Tele-operar un manipulador robótico a través de internet.

6.3.1.2 Necesidades. Hay que leer de nuevo las necesidades del cliente para evitar perder de vista el objetivo real que trae consigo la satisfacción del cliente con un producto a la medida... Ver cuadro 10...

6.3.1.3 Especificaciones. Es importante prestar atención a las especificaciones preliminares para cumplir en lo posible con los valores ideales... Ver cuadro 17...

6.3.2 Descomposición del problema. Se comienza desde lo general para ir hacia lo particular, es decir, se habla en términos generales del producto y sus funciones para ir poco a poco introduciéndose al interior de este y descubrir que lo compone. Para ello se tienen dos herramientas: la vista como caja negra y la descomposición funcional.

6.3.2.1 Caja Negra. Se imagina como si el producto fuese una caja negra, en donde, se desconoce qué compone al sistema. Se sabe con certeza que el sistema necesita unas entradas para reaccionar y producir unas salidas. En la figura 12 se ilustra lo anterior.

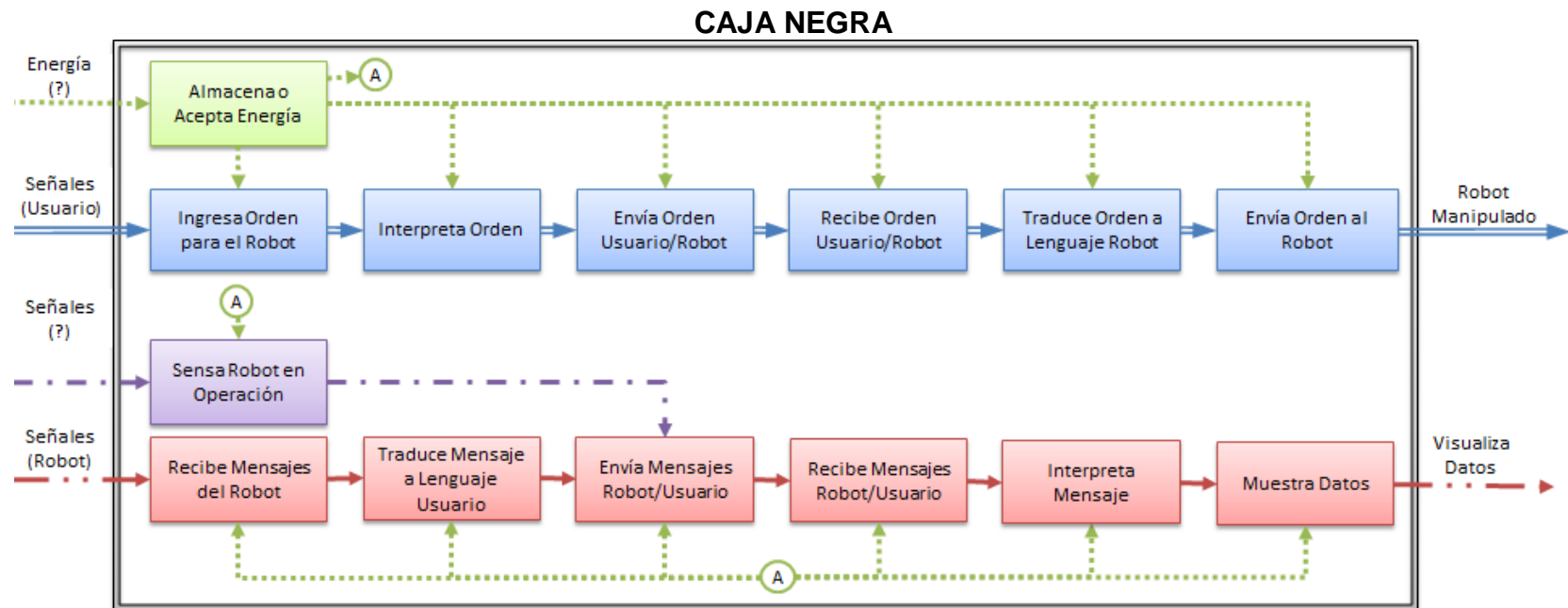
Figura 12. Vista del proyecto como caja negra



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.2.2 Descomposición funcional. Se descompone la caja negra en funciones ampliando o escudriñando su interior para dar solución a cada una de las funciones con las tecnologías disponibles, generando conceptos o ideas. En la figura 13 se hace lo antedicho.

Figura 13. Descomposición funcional



6.3.3 Búsqueda externa e interna de conceptos. En esta etapa se hacen encuestas a especialistas y/o clientes, se hace benchmarking de productos en el mercado, o ambas. Todo con el objetivo de recoger la mayor cantidad de ideas o conceptos que satisfagan cada función del sistema, incluso cuando aquellas ideas parezcan descabelladas, imposibles, fantasiosas o costosas.

Más adelante, se unen las diferentes alternativas para crear posibles soluciones a la problemática del proyecto. Las opciones que NO pasaron las condiciones de diseño se guardan para el futuro en el que su desarrollo NO sea una locura.

6.3.3.1 Subfunción almacena o acepta energía. Todo sistema natural o artificial funciona con energía. La energía se encuentra naturalmente en muchos estados diferentes, simplemente hay que transformarla al estado que se necesita o utilizarla sin procesarla. El cuadro 18 muestra esas posibles opciones para obtener la energía que necesita el sistema.

Cuadro 18. Subfunción A: almacena o acepta energía

#	Concepto
A1	Energía eléctrica suministrada del tomacorriente.
A2	Energía eléctrica suministrada a través de baterías.
A3	Energía lumínica transformada a eléctrica obtenida de Celdas Solares.
A4	Energía cinética transformada a eléctrica generada por el movimiento del aire (Eólica).
A5	Energía cinética transformada a eléctrica generada por el movimiento del agua (Hidráulica).
A6	Energía cinética transformada a eléctrica generada por la combustión de mezcla aire-combustible (Butano, Propano, etc.) o por la explosión de polvora que mueve unos pistones.
A7	Energía Geotérmica.
A8	Potencia humana de brazos o piernas. Se transforma la energía transmitida por la fuerza a través de manivelas o caminadoras.
A9	Energía nuclear.

6.3.3.2 Subfunción ingresa orden para el robot. Aquí es donde el usuario interactúa físicamente con un dispositivo intermediario o directamente con el robot para darle los comandos a realizar, en el anexo J se dispone de los comandos que usa el robot. En el cuadro 19 se ilustran los conceptos para esta subfunción.

Cuadro 19. Subfunción B: ingresa orden para el robot

#	Concepto
B1	Digitar comandos en un teclado QWERTY para computador.
B2	Digitar comandos en un teclado para celular.
B3	Digitar comandos en un teach pendant.
B4	Suministrar los comandos por voz a través de un micrófono.
B5	Ingresa los comandos por imágenes a través de una cámara, cada comando será representado por una imagen en particular.
B6	Los comandos están representados por botones virtuales que se activan con el puntero del mouse. Adicionalmente, cada GDL del robot puede estar asociado a los botones físicos del mouse.
B7	Por Joystick. Cada botón tiene asignado un sentido de movimiento de un GDL específico.
B8	Digitar los comandos a través de una pantalla touch. Se hace una interfaz gráfica con todos los comandos asignados a botones virtuales.
B9	Digitar los comandos en un teclado Braille.

6.3.3.3 Subfunción interpreta orden. Los comandos dados por el usuario son procesados para posteriormente enviarlos al robot. El cuadro 20 muestra las ideas que solucionan la subfunción interpreta orden.

Cuadro 20. Subfunción C: interpreta orden

#	Concepto
C1	CPU de un PC interpreta la subfunción ingresa orden para el robot.
C2	Se programa un Microcontrolador para que realice todos los procesos necesarios.
C3	CPU del celular interpreta la subfunción ingresa orden para el robot.

6.3.3.4 Subfunción envía orden usuario/robot. La información es enviada desde donde se encuentra el usuario hasta donde se encuentra el robot. Se puede apreciar las ideas para esta subfunción en el cuadro 21.

Cuadro 21. Subfunción D: envía orden usuario/robot

#	Concepto
D1	Vía Cliente TCP/IP Java
D2	Vía telefónica. Usando un teléfono fijo que envía los comandos dados por voz.
D3	Vía telefónica. Usando un celular que envía los comandos por mensajes de texto.
D4	Vía telefónica. Usando un celular que envía los comandos dados por voz.
D5	Vía Cliente RTP Java

#	Concepto
D6	Vía Cliente UDP Java
D7	Vía Telnet

En el anexo K se listan los formatos media soportados por la librería Java Media Framework (JMF) 2.1.1, en el anexo L los formatos soportados para transmitir via RTP y en el anexo M los dispositivos de captura soportados por la librería.

6.3.3.5 Subfunción recibe orden usuario/robot. Los datos enviados desde el usuario son recibidos para pasarlos al robot. El cuadro 22 enumera las ideas generadas para esta subfunción.

Cuadro 22. Subfunción E: recibe orden usuario/robot

#	Concepto
E1	Vía Servidor TCP/IP Java
E2	Vía telefónica. Usando un teléfono fijo que recibe los comandos dados por voz y son interpretados para enviarlos al robot.
E3	Vía telefónica. Usando un celular que recibe los comandos por mensajes de texto.
E4	Vía telefónica. Usando un celular que recibe los comandos dados por voz y son interpretados para enviarlos al robot.
E5	Vía Servidor RTP Java
E6	Vía Servidor UDP Java
E7	Vía Telnet

6.3.3.6 Subfunción traduce orden a lenguaje robot. Esta subfunción F ya está solucionada en el diseño del robot. Los comandos se envían y el dispositivo los interpreta automáticamente a través del controlador propio.

6.3.3.7 Subfunción envía orden al Robot. El robot fue diseñado para comunicarse con el protocolo RS232C, por ello, la tecnología queda sujeta para el uso de este protocolo específico. El cuadro 23 recoge las ideas que solucionan esta subfunción. En el anexo N se adjunta la Licencia de uso de la librería RXTX.

Cuadro 23. Subfunción G: envía orden al Robot

#	Concepto
G1	API Giovynet para Java. API que permite usar el puerto serial con Java.

#	Concepto
G2	API Java Comm. API que permite usar el puerto serial con Java.
G3	API RXTX para Java. API que permite usar el puerto serial con Java.
G4	Hyperterminal. Programa de Windows que permite comunicaciones seriales.
G5	Advanced Terminal Software – ATS. Software diseñado para trabajar con el robot usando el lenguaje propio.

6.3.3.8 Subfunción sensa robot en operación. Es necesario que el usuario tenga conocimiento de qué está pasando con el robot en todo momento, por ejemplo, cuando le envía comandos; esto es para confirmar si realizó la operación deseada. En el cuadro 24 se enuncia las posibles formas de sensar el robot en operación.

Cuadro 24. Subfunción H: sensa robot en operación

#	Concepto
H1	Cámara Web. Se envían imágenes obtenidas de una cámara hacia el lugar de control a través de internet.
H2	Cámara. Se envían imágenes obtenidas de una cámara hacia el lugar de control a través de un canal televisivo.
H3	Simulación. Se asocian las posiciones obtenidas a una imagen que simula las posiciones del robot.

6.3.3.9 Subfunción recepción de mensajes del robot. Al igual que la subfunción envía orden al robot, esta está sujeta a la tecnología RS232. En el cuadro 25 se presentan las soluciones para esta subfunción.

Cuadro 25. Subfunción I: recepción de mensajes del robot

#	Concepto
I1	API Giovynet para Java. API que permite usar el puerto serial con Java.
I2	API Java Comm. API que permite usar el puerto serial con Java.
I3	API RXTX para Java. API que permite usar el puerto serial con Java.
I4	Hyperterminal. Programa de Windows que permite comunicaciones seriales.

#	Concepto
I5	Advanced Terminal Software – ATS. Software diseñado para trabajar con el robot usando el lenguaje propio ACLwin.

6.3.3.10 Subfunción traduce mensaje a lenguaje usuario. Esta subfunción J ya está solucionada en el diseño del robot. Los mensajes del robot se envían y el dispositivo los traduce al inglés automáticamente a través del controlador propio.

6.3.3.11 Subfunción envía datos robot/usuario. Cuando el usuario envía un comando, el robot responde con un mensaje dando a conocer si el comando fue ejecutado o si es un comando erróneo, es necesario enviarle este mensaje al usuario. Las ideas para realizar esto, se presentan en el cuadro 26.

Cuadro 26. Subfunción K: envía datos robot/usuario

#	Concepto
K1	Vía Servidor TCP/IP Java
K2	Vía telefónica. Usando un teléfono fijo, en donde, el robot es adaptado para responder a los comandos dados por voz, por medio de palabras audibles pregrabadas.
K3	Vía telefónica. Usando un celular que envía la respuesta del robot por mensajes de texto.
K4	Vía telefónica. Usando un celular, en donde, el robot es adaptado para responder a los comandos dados por voz, por medio de palabras audibles pregrabadas.
K5	Vía Servidor RTP Java
K6	Vía Servidor UDP Java
K7	Vía Telnet

6.3.3.12 Subfunción recibe datos robot/usuario. Esta función es la encargada de recibir los mensajes transmitidos del robot. Las posibles soluciones se encuentran en el cuadro 27.

Cuadro 27. Subfunción L: recibe datos robot/usuario

#	Concepto
L1	Vía Cliente TCP/IP Java
L2	Vía telefónica. Usando un teléfono fijo que recibe los mensajes audibles del robot.
L3	Vía telefónica. Usando un celular que recibe los mensajes del robot por mensajes de texto.
L4	Vía telefónica. Usando un celular que recibe los mensajes audibles del robot.
L5	Vía Cliente RTP Java

#	Concepto
L6	Vía Cliente UDP Java
L7	Vía Telnet

6.3.3.13 Subfunción interpreta mensaje. La información enviada desde el robot es procesada para presentarla al usuario. En el cuadro 28 se dispone de las soluciones para el concepto interpreta mensaje.

Cuadro 28. Subfunción M: interpreta mensaje

#	Concepto
M1	CPU de un PC realiza la subfunción interpreta mensaje.
M2	Se programa un Microcontrolador para que realice todos los procesos necesarios.
M3	CPU del celular realiza la subfunción interpreta mensaje.

6.3.3.14 Subfunción muestra datos. Los datos transmitidos desde el robot son mostrados al usuario a través de algún dispositivo. El cuadro 29 ilustra los conceptos para esta subfunción.

Cuadro 29. Subfunción N: muestra datos

#	Concepto
N1	La información es presentada en el monitor de un computador.
N2	La información es presentada en la pantalla de un celular.
N3	La información es presentada en un holograma.
N4	La información es presentada en un Video proyector.
N5	La información es presentada en la pantalla de un Teach Pendant.
N6	La información es presentada en un dispositivo que interpreta hacia Braille.

6.3.4 Conceptos. En la anterior sección se generaron diferentes ideas para solucionar las subfunciones presentadas en la caja negra. En esta sección se unen las ideas para crear conceptos que solucionan el problema abordado en este proyecto.

6.3.4.1 Concepto 1: Tele-operar por Computador. La energía es suministrada del tomacorriente. Se ingresan los comandos a través de un teclado Qwerty para un computador que realiza el procesamiento. Se envían los comandos desde la

CPU local por medio de un cliente TCP/IP usando Java. Estos se reciben en un servidor TCP/IP que usa la misma tecnología del cliente y son enviados al puerto serie usando la librería RXTX para Java, mientras el robot los interpreta. Una cámara web registra los movimientos del manipulador para que el usuario pueda verlos. Los mensajes enviados al puerto serie que provienen del robot son tratados por la librería RXTX. El controlador del brazo robótico traduce los mensajes al inglés y se envían desde el servidor TCP/IP mencionado. El video se envía desde un servidor RTP y se recibe desde un cliente RTP usando JMF. Los mensajes son recibidos por el cliente TCP/IP mencionado. La información es procesada por el computador mencionado inicialmente y se muestra en el monitor. La ecuación 4 muestra qué subfunciones componen el concepto 1. Las ventajas y desventajas de desarrollar el concepto actual se exponen en el cuadro 30 y en la figura 14 se ilustra el presente concepto.

Ecuación 4. Subfunciones que componen el concepto 1

$$A_1 + B_1 + C_1 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_1 = 1$$

Cuadro 30. Ventajas y desventajas del concepto 1

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	---
2	El computador es relativamente fácil de conseguir para el usuario y es útil para otras cosas.	---
3	El internet (TCP/IP) es ampliamente utilizado.	---
4	Las librerías son de uso libre.	---
5	Es muy económico.	---
6	La universidad posee el hardware y software mencionado.	---

Figura 14. Imagen del concepto 1



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.2 Concepto 2: Tele-operar por Teach Pendant. Todas las funciones son iguales a las del concepto 1 excepto porque se ingresan los comandos a través de un Teach Pendant y los mensajes del robot son mostrados en este. El monitor (N_1) se utiliza para mostrar el video que sensa los movimientos del robot. La ecuación 5 muestra qué subfunciones componen el concepto 2, la figura 15 representa el concepto actual y, sus ventajas y desventajas se exponen en el cuadro 31.

Ecuación 5. Subfunciones que componen el concepto 2

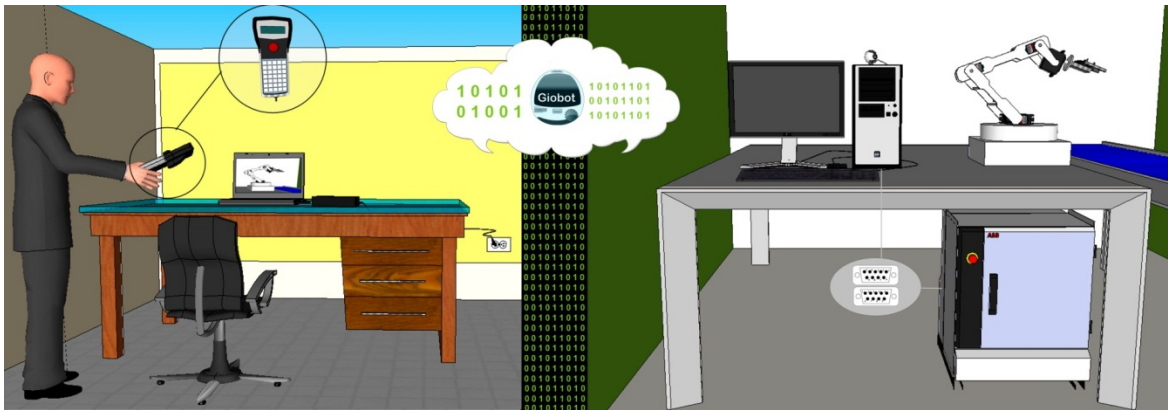
$$A_1 + B_3 + C_1 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_1 + N_5 = 2$$

Cuadro 31. Ventajas y desventajas del concepto 2

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Todos los usuarios tendrían que comprar el Teach Pendant generando costos adicionales.
2	El computador es asequible para el usuario y es útil para otras cosas.	Después de las prácticas al usuario NO le sería útil el Teach Pendant.
3	El internet (TCP/IP) es ampliamente utilizado.	Al usar el Teach Pendant se consume energía adicional, generando costos.
4	Las librerías son de uso libre.	---
5	Es económico.	---

#	Ventajas	Desventajas
6	La universidad posee el hardware y software mencionado.	---

Figura 15. Imagen del concepto 2



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.3 Concepto 3: Tele-operar por Teléfono fijo. La energía es suministrada del tomacorriente. Los comandos son dados por voz a través de un micrófono (del teléfono) y se envían por la red telefónica. Estos se reciben usando el mismo medio y se conectan a un computador, el cual, realiza el procesamiento. Los comandos recibidos se envían al puerto serie usando la librería RXTX para Java y el robot los interpreta. Una cámara web registra los movimientos del robot para que el usuario pueda verlos. Los mensajes enviados al puerto serie que provienen del robot son tratados por la librería RXTX después de que el controlador del robot los haya traducido al inglés. Estos son procesados, codificados para ser escuchados y enviados por vía telefónica para ser recibidos en forma de voz. El video se envía desde un servidor RTP usando JMF y se recibe desde un cliente que usa la misma tecnología para ser mostrado en el monitor del computador. La ecuación 6 muestra qué subfunciones componen el concepto 3. El cuadro 32 presenta las ventajas y desventajas del concepto actual. En la figura 16 se muestra el concepto 3.

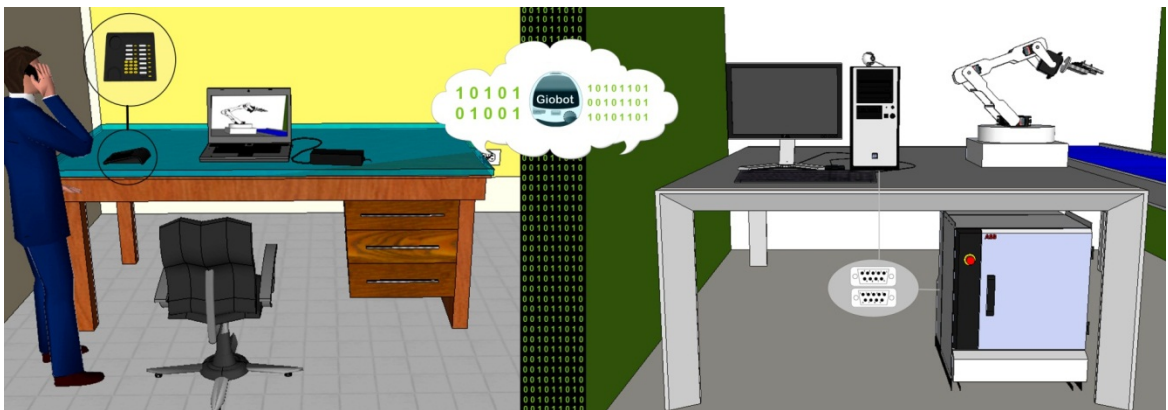
Ecuación 6. Subfunciones que componen el concepto 3

$$A_1 + B_4 + C_1 + D_2 + E_2 + F + G_3 + H_1 + I_3 + J + K_2 + K_5 + L_2 + L_5 + M_1 + N_1 = 3$$

Cuadro 32. Ventajas y desventajas del concepto 3

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	No se aprovecha el computador al presentar solamente las imágenes de la cámara.
2	El computador es relativamente fácil de conseguir y es útil para otras cosas.	---
3	El internet (TCP/IP) es ampliamente utilizado.	---
4	Las librerías son de uso libre.	---
5	Es económico.	---
6	El teléfono se encuentra en la mayoría de hogares e instituciones.	---
7	La universidad posee el hardware y software mencionado.	---

Figura 16. Imagen del concepto 3



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.4 Concepto 4: Tele-operar usando Procesamiento de imágenes. Este concepto presenta la mayoría de las subfunciones del concepto 1, la excepción es la forma de ingresar los comandos, se hacen señas o se presentan imágenes delante de la cámara para representar los comandos. La ecuación 7 muestra qué subfunciones componen el concepto 4, el cuadro 33 presenta las ventajas y desventajas, y en seguida, se muestra en la figura 17.

Ecuación 7. Subfunciones que componen el concepto 4

$$A_1 + B_4 + C_1 + D_5 + E_5 + F + G_3 + H_1 + I_3 + J + K_5 + L_5 + M_1 + N_1 = 4$$

Cuadro 33. Ventajas y desventajas del concepto 4

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	El usuario necesita adquirir cámaras especializadas, generando un costo adicional.
2	El computador es asequible para el usuario y es útil para otras cosas.	Se necesita más trabajo e investigación para el procesamiento de imágenes.
3	El internet (TCP/IP) es ampliamente utilizado.	Ocupa más espacio en disco.
4	Las librerías son de uso libre.	---
5	La universidad posee el hardware y software mencionado.	---

Figura 17. Imagen del concepto 4



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.5 Concepto 5: Tele-operar con Mouse. Al igual que el anterior concepto, la mayoría de subfunciones son solucionadas de igual manera que el concepto 1, lo único que los diferencia, es la forma de ingresar los comandos, en el presente concepto se asocian los comandos a botones virtuales, evitando la posibilidad de que el usuario olvide usar algunos comandos y le saque el mejor provecho,

complementariamente, los botones físicos del mouse se pueden asociar a los GDL del robot, permitiendo moverlo libremente con el mouse. En la ecuación 8 se representa el presente concepto, en seguida, el cuadro 34 presenta sus ventajas y desventajas; y la figura 18 lo expone.

Ecuación 8. Subfunciones que componen el concepto 5

$$A_1 + B_6 + C_1 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_1 = 5$$

Cuadro 34. Ventajas y desventajas del concepto 5

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	---
2	El computador es asequible para el usuario y es útil para otras cosas.	---
3	El internet (TCP/IP) es ampliamente utilizado.	---
4	Las librerías son de uso libre.	---
5	Es muy económico.	---
6	Todos los comandos son tenidos en cuenta por el usuario.	---
7	La universidad posee el hardware y software mencionado.	---

Figura 18. Imagen del concepto 5



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.6 Concepto 6: Tele-operar con Joystick. Es un derivado del concepto 1, con la diferencia de que el robot es tele-operado por un Joystick. La idea es asignar los movimientos o GDL a los botones del Joystick. Las subfunciones que componen este concepto se aprecian en la ecuación 9. En la figura 19 se muestra el concepto 6, pero antes, el cuadro 35 presenta sus ventajas y desventajas.

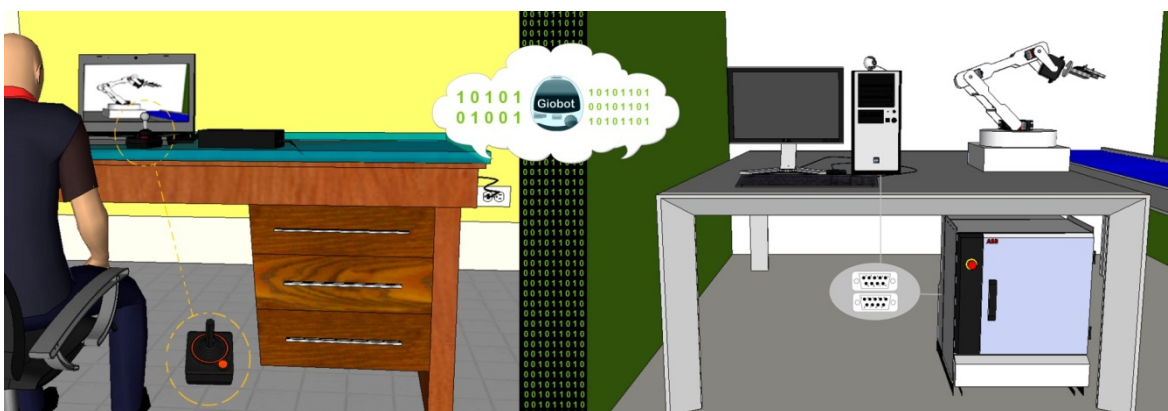
Ecuación 9. Subfunciones que componen el concepto 6

$$A_1 + B_7 + C_1 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_1 = 6$$

Cuadro 35. Ventajas y desventajas del concepto 6

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Que el usuario compre un Joystick, aumentándole costos.
2	El computador es asequible para el usuario y es útil para otras cosas.	Posiblemente después de la práctica, el Joystick NO se vuelve a usar.
3	El internet (TCP/IP) es ampliamente utilizado.	---
4	Las librerías son de uso libre.	---
5	Es económico.	---
6	La universidad posee el hardware y software mencionado.	---

Figura 19. Imagen del concepto 6



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.7 Concepto 7: Tele-operar con Pantalla Táctil. Este es otro concepto derivado del 1 unido al principio del concepto 5, en este los comandos son asignados a botones virtuales, los cuales se activan por las pulsaciones sobre una pantalla táctil. La ecuación 10 presenta la unión de varias subfunciones que definen el concepto 7. En la figura 20 se muestra el concepto seguido del cuadro 36 que presenta sus ventajas y desventajas.

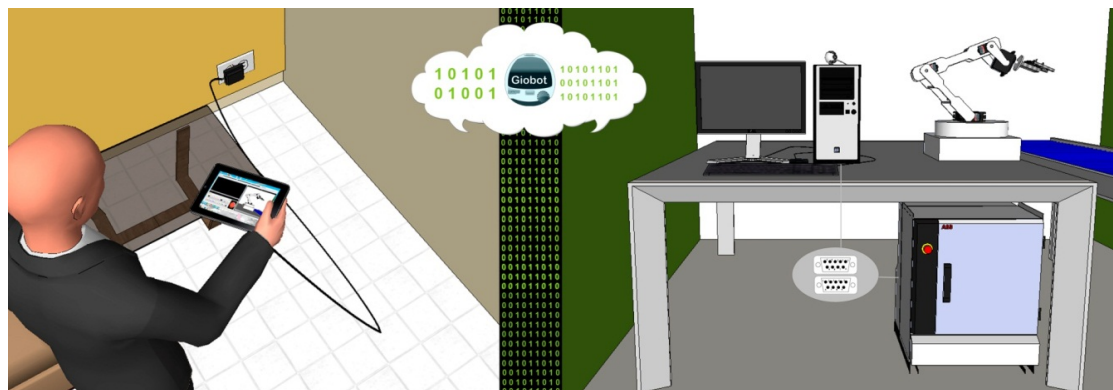
Ecuación 10. Subfunciones que componen el concepto 7

$$A_1 + B_8 + C_1 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_1 = 7$$

Cuadro 36. Ventajas y desventajas del concepto 7

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	El usuario tiene que adquirir una pantalla táctil, la cual NO es una necesidad, generando costos.
2	El computador es asequible para el usuario y es útil para otras cosas.	---
3	El internet (TCP/IP) es ampliamente utilizado.	---
4	Las librerías son de uso libre.	---
5	Todos los comandos son tenidos en cuenta por el usuario.	---
6	La universidad posee el hardware y software mencionado.	---

Figura 20. Imagen del concepto 7



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.8 Concepto 8: Tele-operar por Celular. El celular recibe la energía de una batería recargable en un tomacorriente. Se ingresan los comandos a través del teclado qwerty o numérico para un celular que realice el procesamiento. Se envían los comandos desde el celular por medio de un cliente TCP/IP Java y se reciben en un computador (o celular) servidor TCP/IP que usa la misma tecnología del cliente. Los comandos recibidos se envían al puerto serie usando la librería RXTX para Java y el robot interpreta los comandos. Una cámara web registra los movimientos del manipulador para que el usuario pueda verlos. Los mensajes del robot se envían al puerto serie usando la librería RXTX mientras el controlador del brazo robótico traduce los mensajes al inglés, entonces, se envían desde el servidor TCP/IP hacia el cliente TCP/IP. El video se envía desde un servidor RTP y se recibe desde un cliente RTP usando JMF. La información es procesada por el celular mencionado inicialmente y se muestra en la pantalla de este. La ecuación 11 define al concepto 8 mientras el cuadro 37 presenta sus ventajas y desventajas. En la figura 21 se muestra el concepto 8.

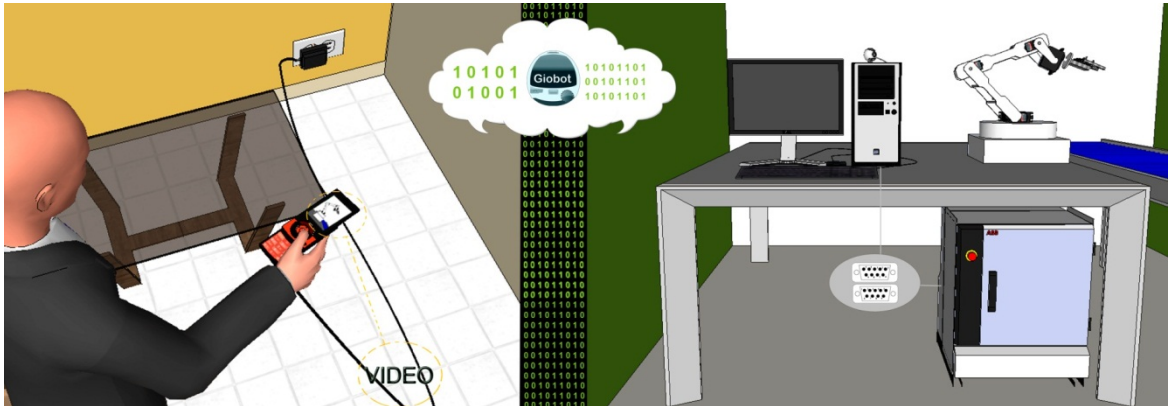
Ecuación 11. Subfunciones que componen el concepto 8

$$A_2 + B_2 + C_3 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_2 = 8$$

Cuadro 37. Ventajas y desventajas del concepto 8

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Pagar una tarifa para internet del celular usado por el usuario, aumentando costos.
2	El computador es asequible para el usuario y es útil para otras cosas.	La señal es muy mala en el Laboratorio de Procesos de Manufactura.
3	El internet (TCP/IP) es ampliamente utilizado.	---
4	Las librerías son de uso libre.	---
5	Es relativamente económico.	---
6	La universidad posee el hardware y software mencionado.	---
7	El celular para el usuario es relativamente fácil de conseguir y es útil para otras cosas.	---

Figura 21. Imagen del concepto 8



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.9 Concepto 9: Tele-operar con Mensajes de Texto. El celular recibe la energía de una batería recargable en un tomacorriente. Se ingresan los comandos a través del teclado qwerty o numérico para un celular que realice el procesamiento. Se envían los comandos desde el celular por medio de mensajes de Texto o SMS y se reciben en un celular que los adapta para ser enviados al puerto serial usando la librería RXTX para Java. El robot interpreta los comandos. Una cámara web registra los movimientos del robot para que el usuario pueda verlos. Los mensajes del robot se envían al puerto serie usando la librería RXTX y el controlador del robot traduce los mensajes al inglés para luego ser enviados desde el celular cercano al robot. El video se envía desde un servidor RTP y se recibe desde un cliente RTP usando JMF. Los mensajes son recibidos y la información es procesada por el celular del usuario y se muestra en la pantalla de este. La ecuación 12 define al concepto 9 y el cuadro 38 presenta sus ventajas y desventajas. En la figura 22 se presenta la imagen del concepto.

Ecuación 12. Subfunciones que componen el concepto 9

$$A_2 + B_2 + C_3 + D_3 + E_3 + F + G_3 + H_1 + I_3 + J + K_3 + K_5 + L_3 + L_5 + M_3 + N_2 = 9$$

Cuadro 38. Ventajas y desventajas del concepto 9

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Adquirir un celular para el lado Servidor, aumentando costos.

#	Ventajas	Desventajas
2	El celular es relativamente fácil de conseguir para el usuario y es útil para otras cosas.	Pagar una tarifa para enviar y recibir SMS, tanto usuario como universidad, aumentando costos.
3	Los mensajes de texto (SMS) son ampliamente utilizados.	La señal es muy mala en el Laboratorio de Procesos de Manufactura.
4	Las librerías son de uso libre.	Los mensajes en ocasiones tardan en llegar o enviarse.
5	Es relativamente económico.	---
6	La universidad posee el hardware y software mencionado.	---

Figura 22. Imagen del concepto 9



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.10 Concepto 10: Tele-operar con Braille. La energía es suministrada del tomacorriente. Se ingresan los comandos a través de un teclado adaptado para Braille. El procesamiento lo hace un computador que envía los comandos por medio de un cliente TCP/IP y se reciben en un servidor TCP/IP usando Java. Los comandos recibidos se envían al puerto serie usando la librería RXTX para Java y el robot interpreta los comandos. Una cámara web registra los movimientos del robot para que el usuario pueda registrarlos. Los mensajes del robot se envían al puerto serie usando la librería RXTX mientras el controlador del robot los traduce al inglés. Los mensajes se envían desde el servidor TCP/IP y son recibidos por el cliente TCP/IP mencionado. El video se envía desde un servidor RTP y se recibe desde un cliente RTP usando JMF. La información es procesada por el computador mencionado inicialmente y se muestra en un dispositivo diseñado para interpretar a Braille. La ecuación 13 muestra qué subfunciones componen el

concepto 10, seguido, el cuadro 39 presenta sus ventajas y desventajas. En la figura 23 se muestra el concepto 10.

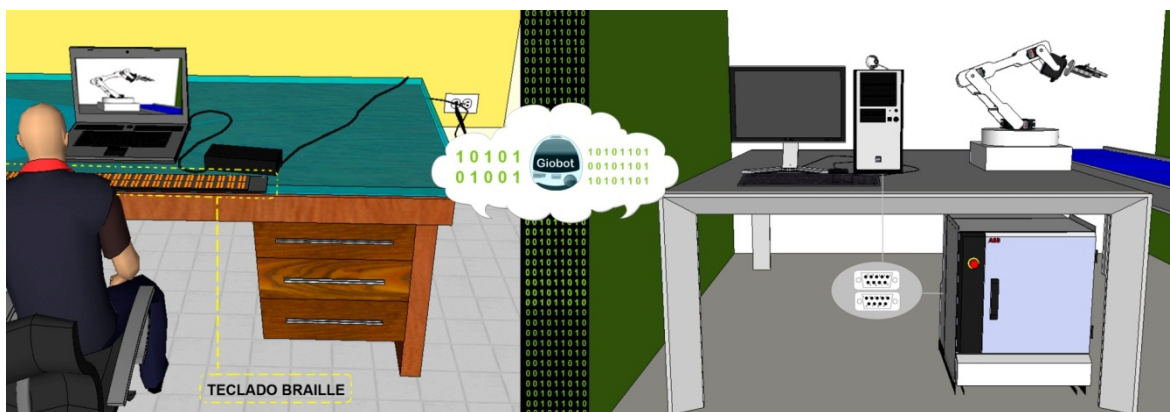
Ecuación 13. Subfunciones que componen el concepto 10

$$A_1 + B_9 + C_1 + D_1 + E_1 + F + G_3 + H_1 + I_3 + J + K_1 + K_5 + L_1 + L_5 + M_1 + N_6 = 10$$

Cuadro 39. Ventajas y desventajas del concepto 10

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Es sólo para personas con poca o visibilidad nula.
2	El computador es asequible para el usuario y es útil para otras cosas.	El usuario tendría que adquirir los dos (2) dispositivos especiales, aumentando costos.
3	El internet (TCP/IP) es ampliamente utilizado.	La universidad no está diseñada para atender personas ciegas.
4	Las librerías son de uso libre.	---
5	Es relativamente económico.	---
6	La universidad posee el hardware y software mencionado.	---
7	Los dispositivos adquiridos por el usuario le serían útiles después de las prácticas.	---

Figura 23. Imagen del concepto 10



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.11 Concepto 11: Tele-operar a través de UDP. La energía es suministrada del tomacorriente. Se ingresan los comandos a través del teclado Qwerty y el procesamiento lo hace un computador que envía los comandos por medio de un cliente UDP y se reciben en un servidor UDP usando Java. Los comandos recibidos se envían al puerto serie usando la librería RXTX para Java mientras el robot los interpreta. Una cámara web registra los movimientos del robot para que el usuario pueda verlos. Los mensajes del robot se envían al puerto serie usando la librería RXTX, entonces, el controlador del robot traduce los mensajes al inglés. Los mensajes se envían desde el servidor UDP y se reciben por el cliente UDP mencionado. El video se envía desde un servidor RTP para ser recibido desde un cliente RTP usando JMF. La información es procesada por el computador mencionado inicialmente y se muestra en el monitor. La ecuación 14 muestra qué subfunciones componen el concepto 11, seguido, el cuadro 40 presenta sus ventajas y desventajas. En la figura 24 se muestra el concepto 11.

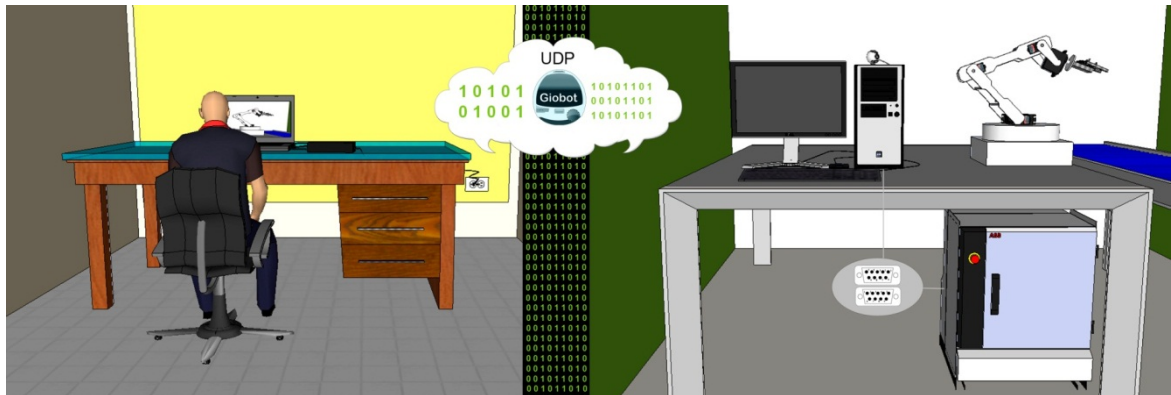
Ecuación 14. Subfunciones que componen el concepto 11

$$A_1 + B_1 + C_1 + D_6 + E_6 + F + G_3 + H_1 + I_3 + J + K_5 + K_6 + L_5 + L_6 + M_1 + N_1 = 11$$

Cuadro 40. Ventajas y desventajas del concepto 11

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Algunos comandos enviados se pierden.
2	El computador es asequible para el usuario y es útil para otras cosas.	Algunos llegan en desorden.
3	El UDP está en todos los computadores.	Algunos comandos son corrompidos en la transmisión.
4	Las librerías son de uso libre.	---
5	Es muy económico.	---
6	La universidad posee el hardware y software mencionado.	---

Figura 24. Imagen del concepto 11



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.12 Concepto 12: Tele-operar a través de Telnet. Su funcionamiento es otro derivado más del concepto 1, con la diferencia de que la información es transmitida por medio del protocolo Telnet. La ecuación 15 muestra qué subfunciones componen el concepto 12. En la figura 25 se muestra el concepto 12, pero antes, el cuadro 41 presenta sus ventajas y desventajas.

Ecuación 15. Subfunciones que componen el concepto 12

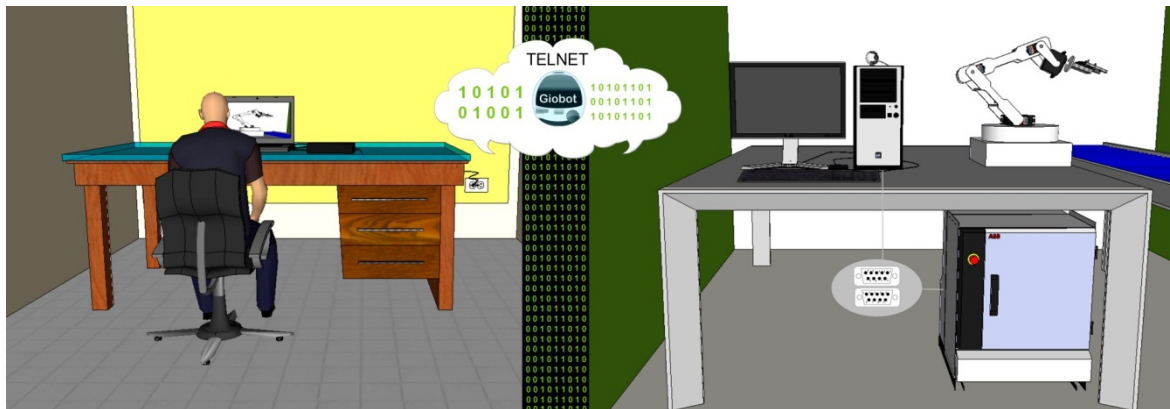
$$A_1 + B_1 + C_1 + D_7 + E_7 + F + G_3 + H_1 + I_3 + J + K_5 + K_7 + L_5 + L_7 + M_1 + N_1 = 12$$

Cuadro 41. Ventajas y desventajas del concepto 12

#	Ventajas	Desventajas
1	La energía está disponible en casi todas partes.	Puede ser espiada la comunicación muy fácilmente.
2	El computador es relativamente fácil de conseguir para el usuario y es útil para otras cosas.	La información puede ser alterada por terceros malintencionados.
3	El Telnet está en todos los computadores.	Terceros malintencionados y con el conocimiento suficiente pueden obtener el nombre de usuario y contraseña.
4	Las librerías son de uso libre.	---
5	Es muy económico.	---

#	Ventajas	Desventajas
6	La universidad posee el hardware y software mencionado.	---

Figura 25. Imagen del concepto 12



Fuente: ZORRILLA PRIETO, Julián Andrés.

6.3.4.13 Otros conceptos. Como es evidente se puede hacer más combinaciones y obtener un mayor número de conceptos, pero NO se realizaron por su inviabilidad debido a compromisos negativos en costos, seguridad, tiempo de diseño e implementación, esfuerzo necesario por el usuario, infraestructura compleja que ocupa mucho espacio o por NO existir la tecnología actualmente. Por ejemplo, se necesita de una infraestructura costosa para que funcionen las siguientes energías: nuclear, eólica, hidráulica, geotérmica, solar, obtenida por combustión de químicos y la energía transferida por la fuerza humana.

6.4 SELECCIÓN DE CONCEPTOS

6.4.1 Tamizaje. Se realiza una matriz, en donde, se tabulan máximo 12 conceptos en las columnas y se tienen en cuenta 10 criterios de selección (necesidades del cliente) para las filas. Luego, se escoge un concepto como referencia, el cual puede ser: un criterio industrial estandarizado para este tipo de producto, el producto mejor evaluado en la sección de Berchmarking, alguno de los conceptos generados, una versión anterior del producto o algunas de las especificaciones preliminares. De acuerdo con lo anterior, el concepto que se toma como referencia es el 1.

El paso a seguir es comparar los conceptos con la referencia y calificarlos, asignando cero (0) cuando es igual, mas (+) cuando es mejor y menos (-) cuando es peor que la referencia.

Finalmente, se totalizan la cantidad de positivos, negativos e iguales, para así, restar el total de positivos con negativos y, con los resultados jerarquizar los conceptos, entonces, se analiza detenidamente si se puede unir algún concepto con otro para mejorar sus características sin hacer demasiados cambios. En el cuadro 42 se combinan los conceptos 1 y 5, y se identifica este acto con la letra U. El fin último de este procedimiento es filtrar conceptos.

Cuadro 42. Tamizaje de Conceptos

#	CRITERIOS DE SELECCIÓN	VARIANTES DE CONCEPTOS												Ref
		1	2	3	4	5	6	7	8	9	10	11	12	
1	Es seguro para el usuario.	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Cumple con las normas técnicas.	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Es seguro para el robot.	0	0	0	0	0	0	0	0	0	0	-	-	0
4	Asegura la operación correcta del robot.	0	0	0	0	+	0	+	0	0	0	-	-	0
5	Puede operar al robot por mando a distancia.	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Tiene Paro de Emergencia.	0	0	0	0	0	0	0	0	0	0	0	0	0
7	La interacción es en tiempo real.	0	0	0	0	0	0	0	0	-	0	-	0	0
8	Se puede visualizar el robot.	0	0	0	0	0	0	0	0	0	-	0	-	0
9	Es económico.	0	-	-	-	0	-	-	+	-	-	0	0	0
10	Permite que el robot pueda estar disponible 24 horas al día.	0	0	0	0	0	0	0	0	0	0	0	0	0
Positivos		0	0	0	0	1	0	1	1	0	0	0	0	
Iguales		10	9	9	9	9	9	8	9	8	8	7	7	
Negativos		0	1	1	1	0	1	1	0	2	2	3	3	
Total (Positivos – Negativos)		0	-1	-1	-1	1	-1	0	1	-2	-2	-3	-3	
Orden		3	8	5	7	1	6	4	2	10	9	12	11	
¿Continuar?		U	NO	NO	NO	U	NO	SI	SI	NO	NO	NO	NO	

6.4.2 Evaluación. Después del Tamizaje o filtrado de conceptos se evalúan aquellos que han pasado a esta etapa para determinar cuál de ellos es el mejor para desarrollar. La calificación se hace asignando los valores expuestos... en el cuadro 3 (Cuadro de Ponderación)... Lo anterior se encuentra organizado en el cuadro 43. El concepto seleccionado es la combinación entre el 1 y el 5, esta unión aumenta la operación correcta del robot mejorando la característica del concepto 1.

Cuadro 43. Evaluación de Conceptos

			VARIANTES DE CONCEPTOS							
			1+5		7		8		Referencia	
#	CRITERIOS DE SELECCION	% Ponderación	Nota	Criterio Ponderado	Nota	Criterio Ponderado	Nota	Criterio Ponderado	Nota	Criterio Ponderado
1	Es seguro para el usuario.	23	3	0,690	3	0,690	3	0,690	3	0,690
2	Cumple con las normas técnicas.	19	3	0,570	3	0,570	3	0,570	3	0,570
3	Es seguro para el robot.	12	3	0,360	3	0,360	3	0,360	3	0,360
4	Asegura la operación correcta del robot.	9	4	0,360	4	0,360	3	0,270	3	0,270
5	Puede operar al robot por mando a distancia.	9	3	0,270	3	0,270	3	0,270	3	0,270
6	Tiene Paro de Emergencia.	9	3	0,270	3	0,270	3	0,270	3	0,270
7	La interacción es en tiempo real.	6	3	0,180	3	0,180	3	0,180	3	0,180
8	Se puede visualizar el robot.	5	3	0,150	3	0,150	3	0,150	3	0,150
9	Es económico.	4	3	0,120	2	0,080	5	0,200	3	0,120
10	Permite que el robot pueda estar disponible 24 horas al día.	4	3	0,120	3	0,120	3	0,120	3	0,120
Total			3,09		3,05		3,08		3,00	
Orden			1		3		2		4	
¿Continuar?			DESARROLLAR		NO		NO		NO	

7. DISEÑO A NIVEL DE SISTEMA

7.1 ARQUITECTURA DEL PRODUCTO

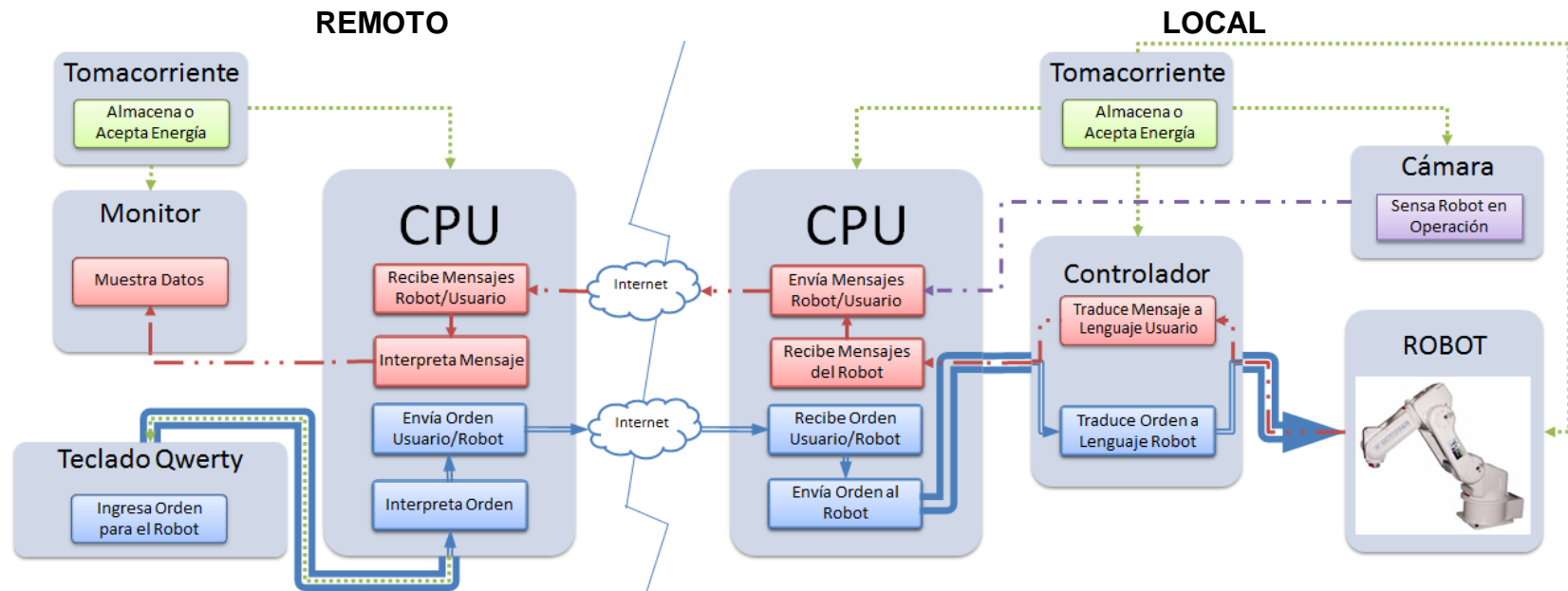
La arquitectura del producto consiste en dividirlo en diferentes partes asignándoles sus respectivas funciones y decidiendo como se conectan. La arquitectura de producto contiene dos ramas: La arquitectura integral y la modular. La integral se caracteriza porque hay un alto grado de interdependencia entre los componentes mientras que la modular se conoce por la estandarización de sus componentes y por ende su alta compatibilidad con otros elementos.

En la sección de generar conceptos, inicialmente se hizo una descomposición funcional, pero no se estableció que dispositivos solucionaban alguna función determinada. En el Diagrama de Despliegue de la figura 26 se puede observar que hay dispositivos que solucionan una función en particular mientras que hay otros que abarcan varias funciones. En términos generales los dispositivos están dispuestos de forma modular.

Pero hay que recordar que el Monitor, la CPU, el teclado, el internet, la cámara, el robot, el controlador y el tomacorriente ya existen, lo que falta por implementar son las funciones que corresponden a la CPU, las cuales son solucionadas por medio de software.

El software usado es Java, un programa con el paradigma Orientado a Objetos, lo cual permite realizar software de forma modular, cuya ventaja es que se pueden reutilizar las partes modulares o se pueden mejorar.

Figura 26. Diagrama de Despliegue

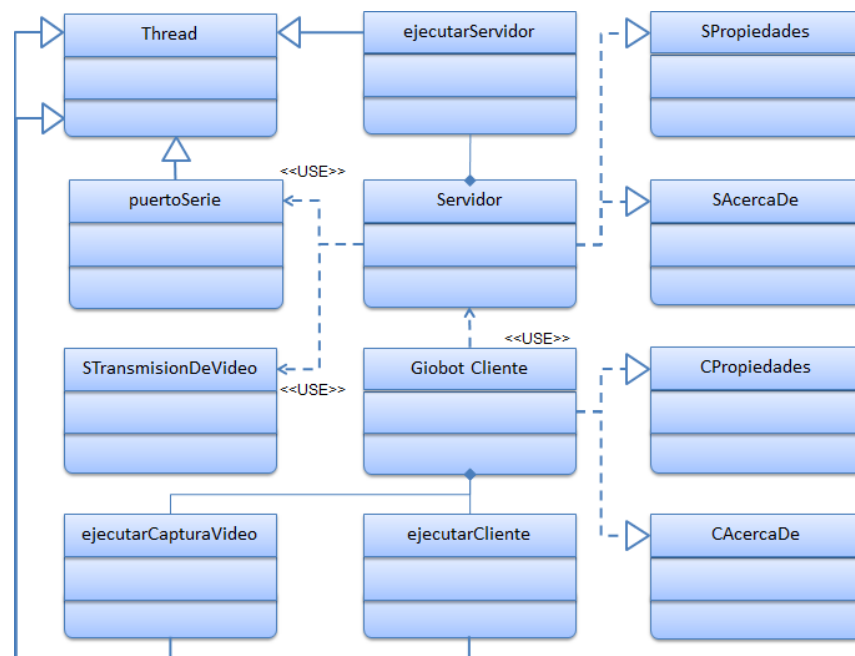


Como se puede ver, las funciones del lado del Cliente son: “Recibe Mensajes Robot/Usuario”, “Interpreta Mensaje”, “Envía Orden Usuario/Robot” e “Interpreta Orden”; y las del lado del Servidor son: “Envía Mensajes Robot/Usuario”, “Recibe Mensajes del Robot”, “Recibe Orden Usuario/Robot” y “Envía Orden al Robot”.

En la figura 27 se desarrolla el Diagrama de Clases de las funciones del lado del Cliente y del Servidor. Las clases ejecutarCapturaVideo y ejecutarCliente estan contenidas en la clase “Giobot Cliente” (ver código en el anexo O), las clases CPropiedades (ver código en el anexo P) y CAcercaDe (ver código en el anexo Q) son usadas por “Giobot Cliente”, la primera es para asignar el puerto TCP de información y el de video; y para establecer la dirección IP a la cual se desea conectar. La segunda clase simplemente suministra información sobre el autor y versión del programa, a su vez, ambas heredan a la clase Thread, que es un hilo de ejecución de Java. Hay una tercer clase, llamada archivosTXT, que no se presenta en el diagrama pero también es usada por “Giobot Cliente” para guardar el historial de comandos ejecutados (ver código en el anexo R).

En el lado del Servidor se encuentra la clase Servidor (ver código en el anexo S) que contiene la clase ejecutarServidor que es un hilo de ejecución, además, la clase Servidor usa la clase puertoSerie, que es un hilo de ejecución encargado de realizar la conexión y procesar todo lo relacionado con el puerto serie (ver código en el anexo T); usa también la clase STransmisionDeVideo (ver código en el anexo U), cuya función es capturar las imágenes de la cámara y enviarlas por la red usando el protocolo RTP; y por último, usa las clases SPropiedades y SAcercaDe, encargadas de proporcionar las características de conexión TCP/IP con RTP y de suministrar la información del autor con la del programa (ver código en el anexo V y en el anexo W), respectivamente.

Figura 27. Diagrama de Clases



La interfaz gráfica de Giobot Cliente se presenta en la figura 28. En la figura 29 se presenta el diálogo de “Propiedades de Conexión” y en la figura 30 está el diálogo “A cerca de...”, ambos pertenecientes a Giobot Cliente. En el cuadro 44 se exponen los botones de Giobot Cliente y sus funciones.

Figura 28. Interfaz Gráfica de Giobot Cliente

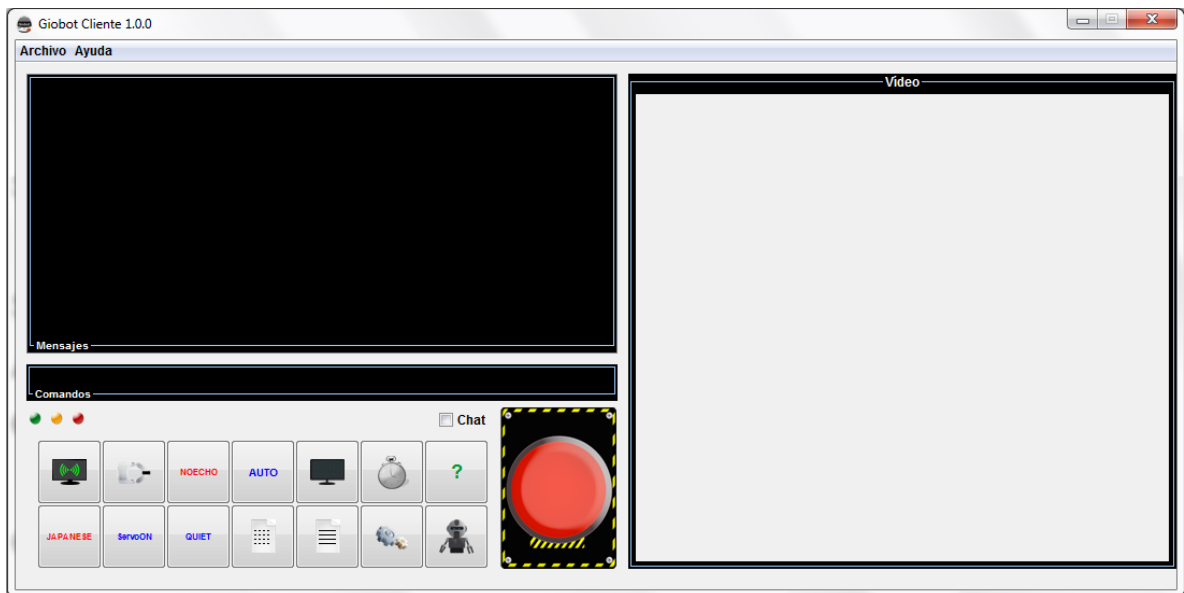


Figura 29. Diálogo CPropiedades

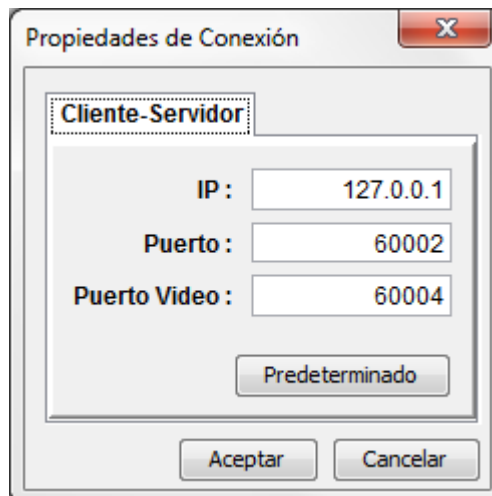


Figura 30. Diálogo CAcercaDe



Cuadro 44. Funciones de los botones de Giobot Cliente



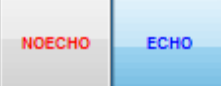


Imagen	Función
	Conectar / Desconectar. Abre un diálogo de propiedades para establecer los parámetros de conexión TCP/IP y RTP, después de establecer la conexión cambia su estado y función para desconectarse de la red.
	Envía el comando "close" y cuando es presionado por segunda vez envía el comando "open".
	Envía el comando "noecho" y cuando es presionado por segunda vez envía el comando "echo".
	Envía el comando "auto".
	Envía el comando "dir".










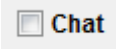
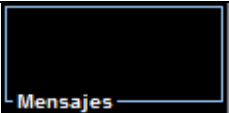
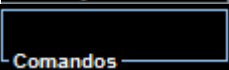
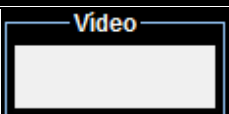

Imagen	Función
	Envía el comando "show speed".
	Envía el comando "help".
	Envía el comando "japanese" y cuando es presionado por segunda vez envía el comando "english".
	Envía el comando "con" y cuando es presionado por segunda vez envía el comando "coff".
	Envía el comando "quiet" y cuando es presionado por segunda vez envía el comando "noquiet".
	Envía el comando "listp".
	Envía el comando "list".
	Envía el comando "config ?".
	Envía el comando "ver".
 Chat	Permite establecer una conversación con alguien que se encuentre del lado del servidor sin enviarla al puerto serie.
 Mensajes	Muestra el historial de comandos y mensajes realizados en la tele-operación.
 Comandos	Lugar donde se escriben los comandos que se enviarán al robot.
 Vídeo	Permite ver al robot en operación si se dispone de una cámara web del lado del servidor.
	Simula un LED que está encendido cuando se ha establecido una conexión con el servidor, de lo contrario, permanece apagado.

Imagen	Función
	Simula un LED que está encendido cuando se está recibiendo el video desde el servidor, de lo contrario, permanece apagado.
	Simula un LED que está encendido cuando se ha oprimido el botón de emergencia, de lo contrario, permanece apagado.
	Envía el comando "a" y cuando es presionado por segunda vez escribe el comando "continue" en el panel de comandos.
	Abre un archivo cuya extensión sea txt y lo muestra en el panel de mensajes o historial. (Ubicado en el menú Archivo).
	Guarda el texto que se encuentra en el panel de mensajes o historial en un archivo txt. (Ubicado en el menú Archivo).
	Borra los mensajes que se encuentran en el panel de mensajes o historial. (Ubicado en el menú Archivo).
	Abre un diálogo de propiedades para establecer los parámetros de conexión TCP/IP y RTP. Esta opción también se encuentra en Giobot Servidor, pero se adiciona la configuración del puerto serie. (Ubicado en el menú Archivo).
	Cierra el programa Giobot Cliente. Esta opción también se encuentra en Giobot Servidor para cerrarlo. (Ubicado en el menú Archivo).
	Abre el documento "Controller-BRC: User manual. Catalog # 100325 Rev. B" en formato pdf. (Ubicado en el menú Ayuda).
	Abre el documento "ACL-Win for Controller-BRC: User manual. Catalog # 100354 Rev. A" en formato pdf. (Ubicado en el menú Ayuda).
	Abre el documento "Motoman XRC: Instruction manual motoman-SV3X, -SV3XL. Manual No. MRS52050" en formato pdf. (Ubicado en el menú Ayuda).
	Abre el documento "Teach Pendant for Controller-BRC: User manual. Catalog # 100353 Rev. A" en formato pdf. (Ubicado en el menú Ayuda).
	Abre un diálogo que muestra la información básica del programa y del autor de este. Esta opción también se encuentra en Giobot Servidor. (Ubicado en el menú Ayuda).

La interfaz gráfica de Giobot Servidor se presenta en la figura 31. La figura 32 ilustra el diálogo SPropiedades y en la figura 33 se expone el diálogo “A cerca de...” correspondientes a Giobot Servidor.

Figura 31. Interfaz Gráfica de Giobot Servidor

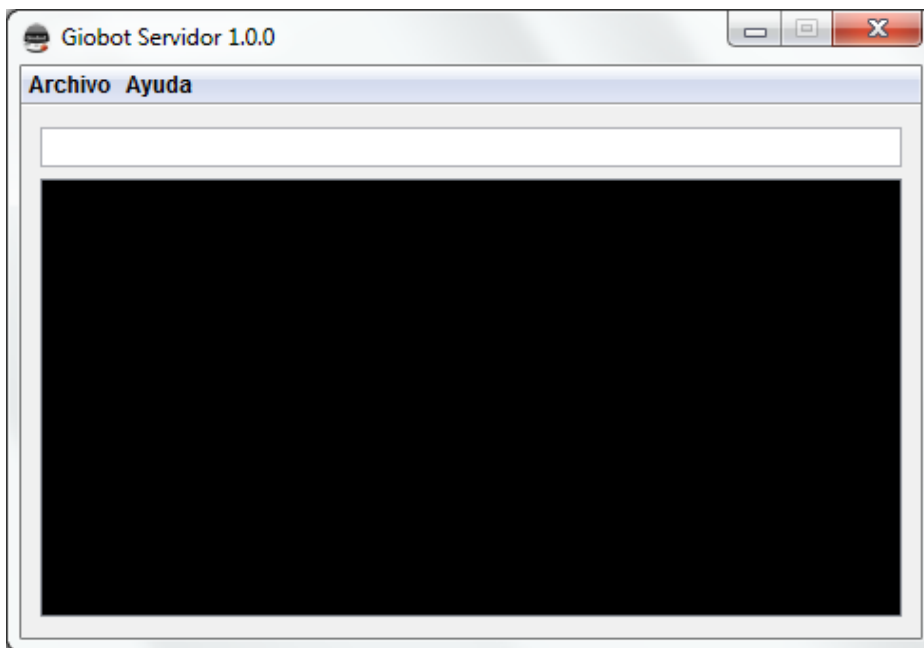


Figura 32. Diálogo SPropiedades

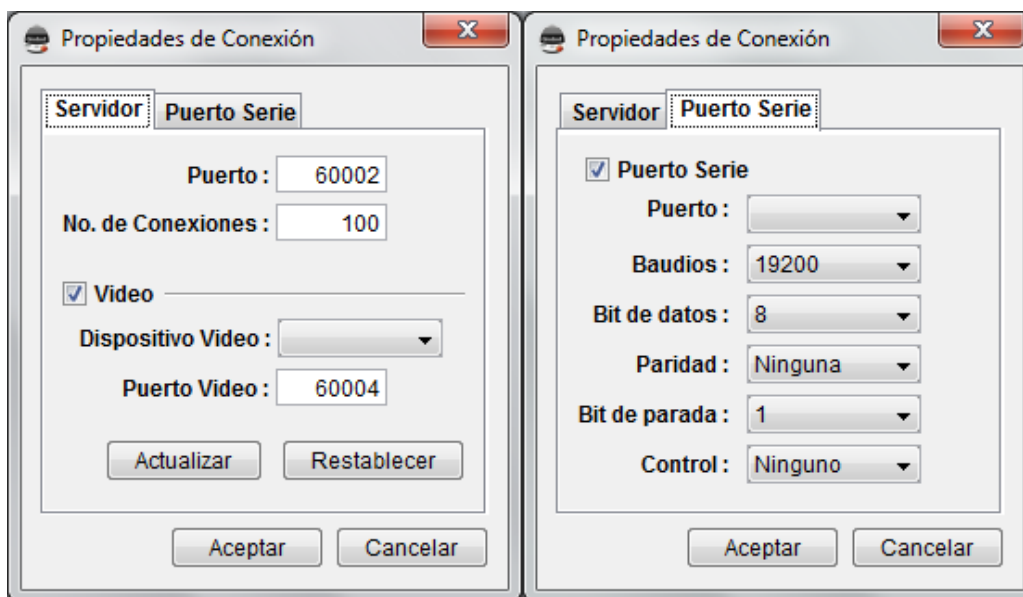


Figura 33. Diálogo SAcercaDe



8. PRUEBAS

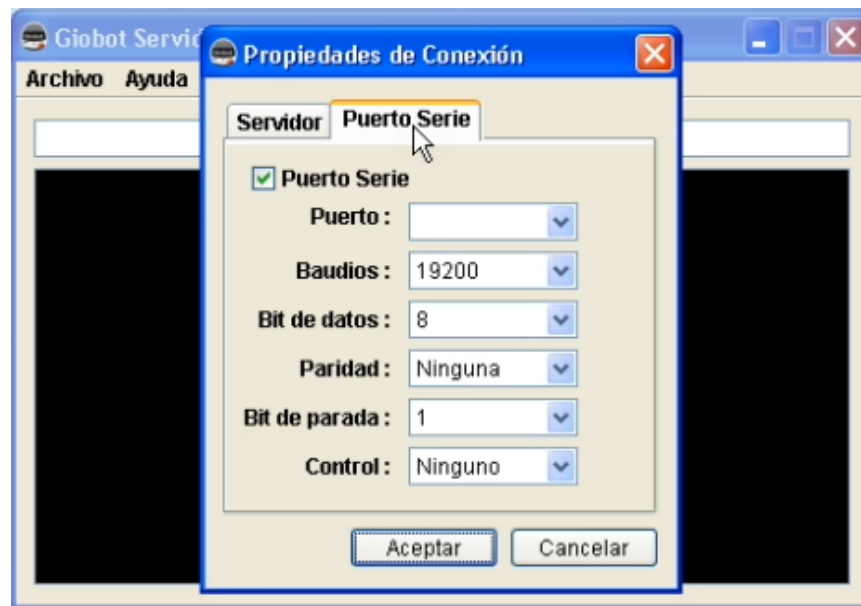
Para que sea posible la ejecución del software, es necesario instalar previamente jdk 1.7, la librería JMF 2.1.1e y librería RXTX, copiar de la carpeta de JMF el archivo JMF.jar y jmf.properties a la carpeta en donde se instaló java en la siguiente ruta “\jdk1.7.0_05\jre\lib\ext\” junto con el archivo RXTXcomm.jar de la librería RXTX, y copiar jmf.properties de JMF, rtxSerial.dll y rtxParallel.dll de RXTX en la ruta “\jdk1.7.0_05\jre\bin” y en “C:\Windows\System32”. Lo anterior es para el sistema operativo Windows, si se va a usar otro sistema operativo consultar la página web de java en la siguiente dirección: <http://www.oracle.com/technetwork/java/javase/download-142937.html> y la de RXTX en: <http://rxtx.qbang.org/wiki/index.php/Installation>.

El siguiente proceso se realizó al interior de la universidad, las pruebas por fuera de la institución no se llevaron a cabo por políticas que protegen a la Autónoma frente a ataques informáticos. Para empezar hay que configurar el Servidor seleccionando el puerto que recibe los comandos del usuario, se establece la cantidad de conexiones esperadas, se escoge el controlador de la cámara, el puerto por donde se transmite el video (ver figura 34) y se define la configuración del puerto serie requerida (ver figura 35), luego se da clic en aceptar para que el Servidor quede a la espera de una conexión.

Figura 34. Configuración de la comunicación desde el Servidor



Figura 35. Configuración del puerto serie



Mientras tanto, en el lado del cliente se da clic en el botón conectar y se establece la dirección IP, puerto para el envío de comandos y el puerto para recibir el vídeo (ver figura 36); una vez se da clic en aceptar, la aplicación queda como se puede observar en la figura 37.

Figura 36. Configuración de la comunicación desde el Cliente

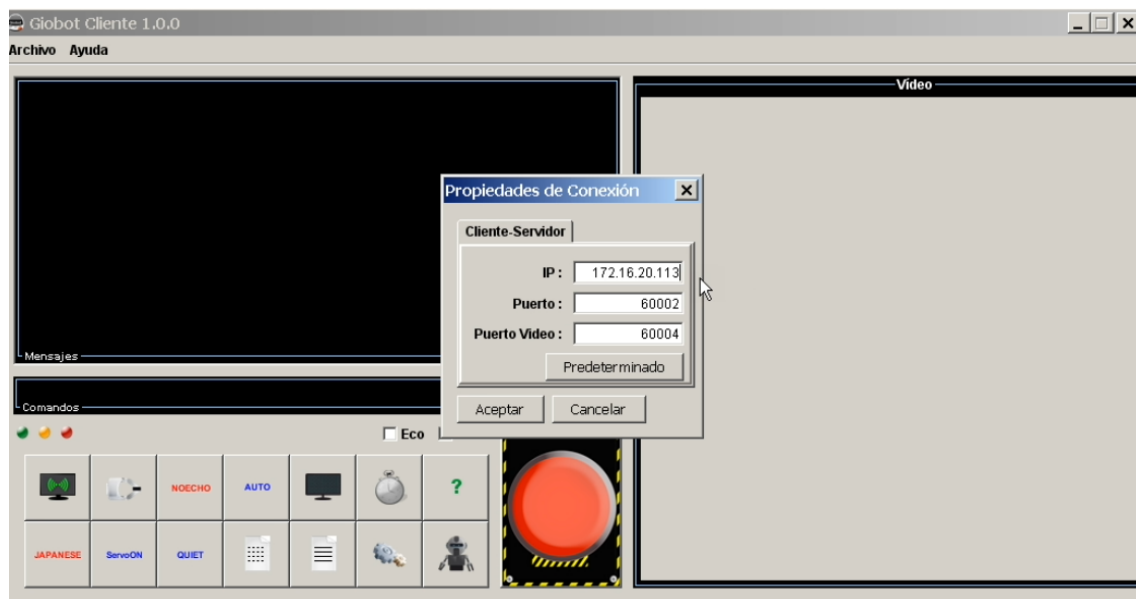
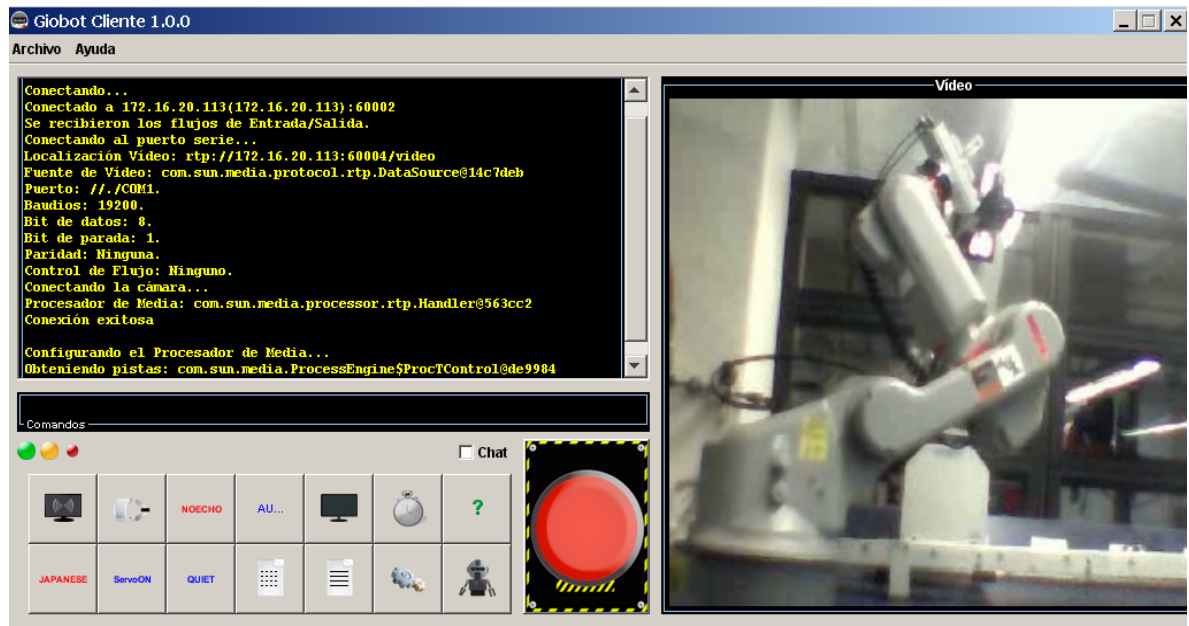
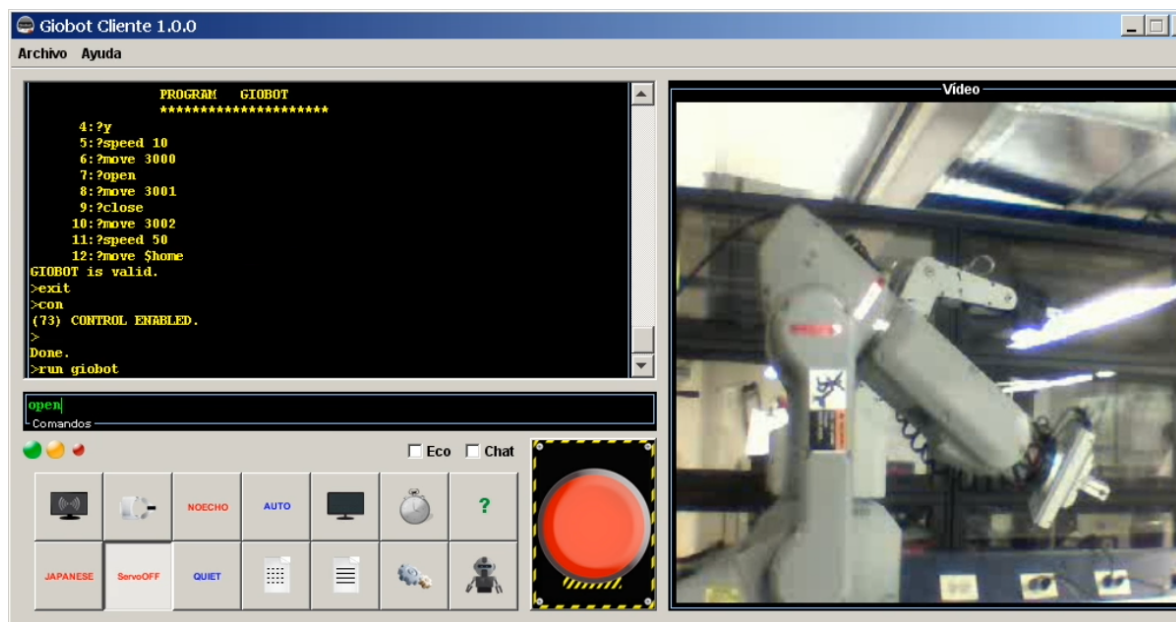


Figura 37. Conexión establecida por el Cliente



Para mover el robot por medio del programa hay que establecer previamente las posiciones que se van a usar a través del Teach Pendant. Primero se mueve el robot a la posición deseada y se da clic en el botón RECORD para luego asignarle un valor numérico a esa posición, después, se crea un programa digitando el comando EDIT junto al nombre del programa y en seguida se escribe la rutina que va a seguir el manipulador finalizando con el comando EXIT. A continuación se muestra paso a paso la creación de la rutina en la figura 38.

Figura 38. Conexión establecida por el Cliente



Para ejecutarla simplemente se escribe RUN Giobot seguido de la pulsación de la tecla enter.

9. CONCLUSIONES

- La cobertura y accesibilidad se mejorarían implementando este proyecto para su uso en dispositivos móviles, ya que, los usuarios disponen del laboratorio desde cualquier lugar.
- La compañía de un tutor en las prácticas remotas asegura el aprendizaje óptimo.
- La comunicación computadora-robot mediante el protocolo RS232 es obsoleta, dado que, esta tecnología ha sido desplazada por el puerto USB que proporciona una velocidad de comunicación superior.
- El diseño del robot dificulta la libertad de manipulación teniendo en cuenta que las posiciones del robot deben ser grabadas previamente en el laboratorio y no es posible hacerlo remotamente.
- La aplicación NO está conectada al botón físico de Paro de Emergencia, lo cual, implica una problema de seguridad latente.

10. RECOMENDACIONES

- Instalar Guardas de Seguridad que delimiten el área de trabajo de los robots.
- Cambiar de lugar los robots a un mueble más pequeño que no presente riesgo de colisión con el robot.
- Capacitar una persona diferente al docente que tenga conocimiento pleno del robot y pueda asesorar a los estudiantes en sus prácticas.
- Realizar el concepto de tele-operación por celular.
- La librería JMF tiene la capacidad de guardar el video capturado, se recomienda implementar esta opción.
- Para mayor compatibilidad de la aplicación con otros robots es necesario crear un diálogo de configuración que permita modificar por el usuario los comandos enviados por los botones virtuales.
- Implementar la conexión de los movimientos de los ejes en modo JOINT a botones virtuales de la aplicación.
- Implementar la conexión de los botones físicos del controlador a botones virtuales de la aplicación.
- Crear el instalador de la aplicación que incluya las librerías.
- Leer la Licencia de la librería RXTX, de la aplicación NetBeans, de la librería Java Media Frameworks (JMF) y los otros productos asociados a Java relacionados con el presente proyecto.
- En ningún caso el autor se hace responsable por cualquier daño o perjuicio, o por cualquier pérdida de ingresos, beneficios o datos, o por daños directos, indirectos, especiales, accidentales o punitivos independientemente de la causa y sin importar la teoría de la responsabilidad, que surja como resultado

del uso, modificación o distribución del contenido de este proyecto o sus derivados, o imposibilidad de uso de lo expuesto, incluso si el autor ha sido advertido de la posibilidad de tales daños.

- En esta sección se pretende diseñar el ambiente de trabajo del robot para realizar prácticas o laboratorios remotos con él. La propuesta que se presenta es usar una de las herramientas adquiridas por la universidad que no está siendo aprovechada, la cual, es la línea de producción dispuesta en el Laboratorio de Procesos de Manufactura. En la figura 39 se aprecia la línea de producción mencionada.

Figura 39. Línea de Producción ubicada en el Laboratorio de Procesos de Manufactura



Esta línea permite que se pueda trabajar con varios robots simultáneamente, por lo cual, se programa una determinada hora para que los estudiantes accedan remotamente a los robots disponibles y trabajen con los robots como si estuviesen fabricando algún producto. Por ejemplo, un estudiante podría programar un robot aumentando la dificultad entre práctica y práctica de la siguiente manera:

- **Primero.** Sujetar con la pinza un marcador y dibujar sobre un panel diferentes figuras geométricas como un triángulo, un cuadrado o un círculo.
- **Segundo.** Desplazar piezas desde una mesa hacia la banda transportadora y viceversa.

- **Tercero.** Seguir con la pinza puntos prefijados sobre la banda transportadora mientras esta se encuentra en movimiento.
- **Cuarto.** Simular la aplicación de soldadura o de pintura sobre una pieza.
- **Quinto.** Desplazar una pieza que se encuentra suspendida en el extremo de una cuerda procurando evitar cualquier movimiento angular.
- **Sexto.** Como trabajo final del curso, se programa cada robot para que uno desplace una pieza desde la mesa hacia la banda transportadora, otro dibuje una figura sobre la pieza mientras esta en movimiento, el siguiente cambie la posición de la figura de horizontal a vertical y finalmente el robot inicial retorne la figura a una mesa.

En la figura 40 se puede analizar la disposición de los robots en la línea de producción y en la figura 41 se encuentra una vista de planta de la anterior figura.

Figura 40. Disposición de los Robots en la Línea de Producción

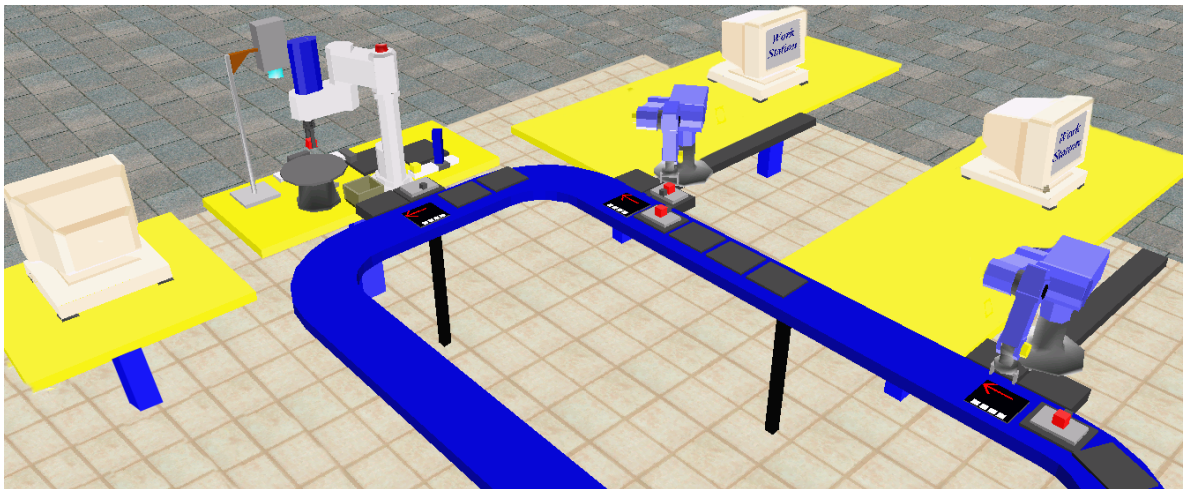
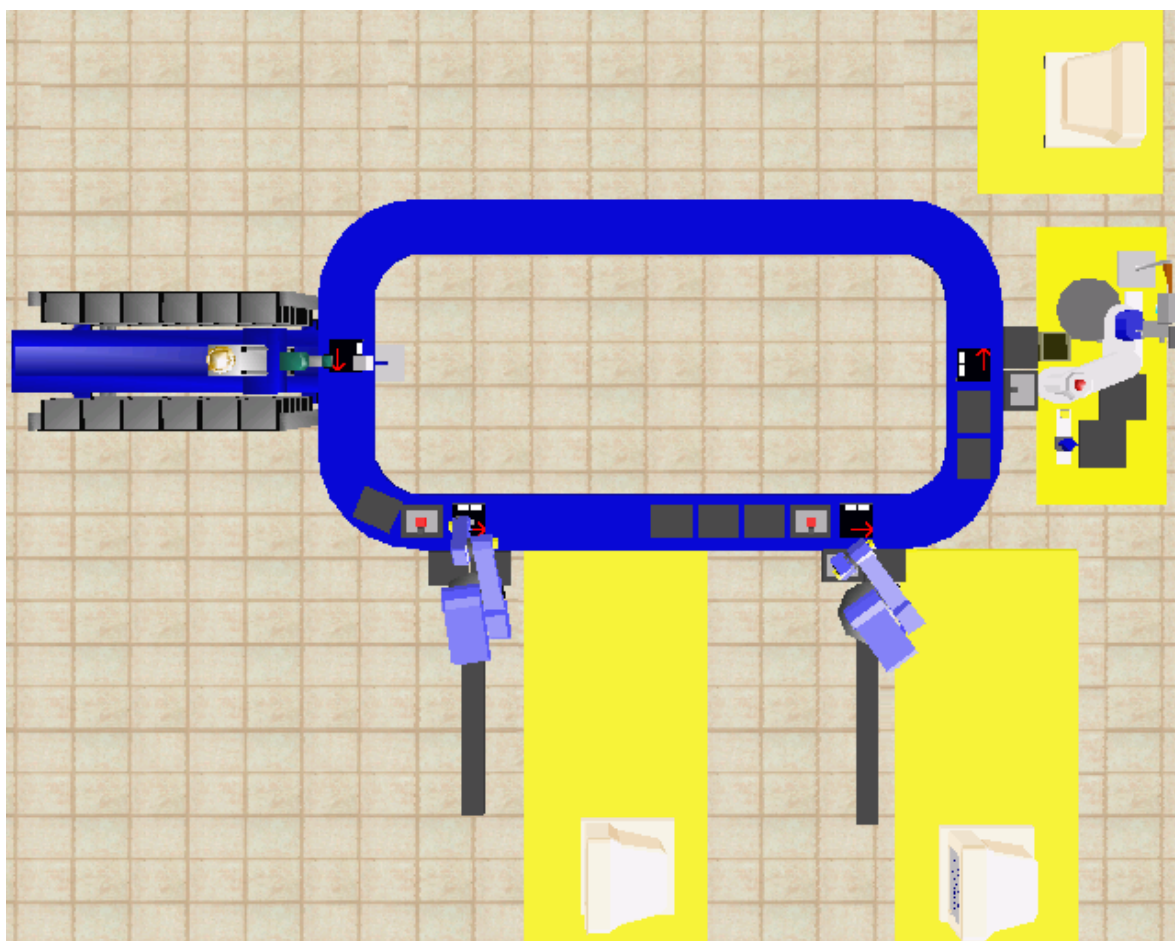


Figura 41. Vista de Planta de la Línea de Producción



BIBLIOGRAFIA

Alegsa. Clasificación de los lenguajes de programación [en línea]. Actualizado 15 de Septiembre de 2006 [Consultado el 16 de Febrero de 2012]. Disponible en internet: <<http://www.alegsaonline.com/art/13.php>>

Asociación de Industrias de Robótica (RIA). Robot Términos y Definiciones [en línea]. Actualizado 19 de Enero del 2010 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://www.robotics.org/product-catalog-detail.cfm?productid=2953>>

Audio y video Reviews. Mando a distancia o control remoto [En línea]. Actualizado 14 de Marzo del 2011 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://audioyvideo.org/index.php?topic=2717.0>>

BARRIENTOS, Antonio; PEÑIN, Luís Felipe y BALAGUER, Carlos. Fundamentos de Robótica. McGraw Hill, 1996. 314 p.

BONAPARTE, Ubaldo. Paradigma Lógico [en línea]. San Miguel de Tucumán, Argentina: Universidad Tecnológica Nacional (UTN). Actualizado 02 de Junio del 2006 [Consultado el 16 de Febrero de 2012]. Disponible en internet: <<http://www.frt.utn.edu.ar/sistemas/paradigmas/page34.html>>

CAICEDO BRAVO, Eduardo., *et al.* Laboratorio Distribuido con Acceso Remoto para la Enseñanza de la Robótica [en línea]. Colombia. Revista Educación en Ingeniería. ISSN 1900-8260. Ed. Asociación Colombiana de Facultades de Ingeniería ACOFI. v.7 fasc.N/A. 11 p, 2009. Disponible en internet: <<http://www.educacioneningenieria.org/index.php/edi/article/view/73/63>>

CAMI Research Inc. The RS232 Standard: Signal Definitions [en línea]. [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>

Departamento de Informática y Automática. Laboratorios virtuales y remotos [en línea]. Madrid, España: Universidad Nacional de Educación a Distancia (UNED). [Consultado el 20 de Marzo de 2011]. Disponible en internet:

<<http://lab.dia.uned.es/rlab/index.html>>

GONZALEZ FERNANDEZ, Víctor R. Robótica Industrial [En línea]. Valladolid, España: Centros de Formación del Profesorado e Innovación Educativa (CFID), 2002. [Consultado el 20 de Marzo de 2011]. Disponible en internet: <http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm>

HUALING, Li. Programming Languages Table—By Capers Jones [en línea]. Actualizado el 05 de Septiembre del 2006 [Consultado el 13 de marzo del 2012]. Disponible en internet: <http://blog.csdn.net/li_hualing/article/details/1179557>

JIMENEZ CHAFLA, Marcela Adriana y ARREGUI JIMENEZ, Gilberth Wilfrido. Diseño e implementación de un sistema de generación de mensajes auditivos para ayudar a personas con discapacidad auditiva y del habla [en línea]. Trabajo de grado Ingeniería en Electrónica y Computación. Riobamba, Ecuador: Escuela Superior Politécnica de Chimborazo, 2009. 177 p. [Consultado el 15 de Enero de 2012]. Disponible en internet: <<http://dspace.esPOCH.edu.ec/bitstream/123456789/182/1/38T00170.pdf>>

LOPEZ FIGUEROLA, Mariano. Teoría de las redes informáticas [en línea]. Actualizado 21 de Julio del 2003 [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://redesafull.galeon.com/#TCP/IP>>

MANZANEDO DEL CAMPO, Miguel Ángel., *et al.* Guía de Iniciación al Lenguaje JAVA: Herencia [en línea]. Burgos, España: Universidad de Burgos, 1999. 189 p. [Consultado el 24 de Febrero de 2012]. Disponible en internet: <<http://zarza.usal.es/~fgarcia/doc/tuto2/Index.htm>>

MARQUES, Pere. Impacto de las TIC en educación: Funciones y limitaciones [en línea]. Barcelona, España: Universidad Autónoma de Barcelona, 2000. [Consultado el 20 de Marzo de 2011]. Disponible en internet: <<http://peremarques.pangea.org/siyedu.htm>>

MICROSOFT. Compatibilidad con herencia de clases en el motor de reglas de negocio [en línea]. [Consultado el 22 de Febrero de 2012]. Disponible en internet: <[http://msdn.microsoft.com/es-es/library/aa560124\(v=bts.10\).aspx](http://msdn.microsoft.com/es-es/library/aa560124(v=bts.10).aspx)>

NATIONAL INSTRUMENTS. Comunicación Serial: Conceptos Generales [en línea]. Actualizado 02 de Enero del 2004 [Consultado el 15 de diciembre del 2011]. Disponible en internet: <<http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>>

OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. 352 p. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

SCHENONE, Marcelo Hernán. Diseño de una Metodología Ágil de Desarrollo de Software [en línea]. Trabajo de grado Ingeniería en Informática. Buenos Aires, Argentina: Universidad de Buenos Aires, 2004. 200 p. [Consultado el 24 de Febrero de 2012]. Disponible en internet: <<http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>>

SEGURIDAD INFORMÁTICA SMR. Tema 6: Seguridad en redes [en línea]. Actualizado 01 de febrero de 2011 [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://seguridadinformaticasmr.wikispaces.com/TEMA+6+-+SEGURIDAD+EN+REDES>>

SUN MICROSYSTEMS. Java TM Media Framework API Guide [en línea]. 1999. 245 p. [Consultado el 08 de Mayo de 2012]. Disponible en internet: <http://java.coe.psu.ac.th/Extension/JMF/jmf2_0-guide.pdf>

The Vision and Autonomous Systems Center (VASC). Release 18 [en línea]. Actualizado el 22 de Febrero del 2005 [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://vasc.ri.cmu.edu/old_help/Misc/Serial/serial.txt>

TIOBE. TIOBE Programming Community Index [en línea]. Actualizado Marzo del 2012 [Consultado el 12 de Marzo de 2012]. Disponible en internet: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>

TOBOSO, Emilio. La Comunicación Serie [en línea]. Actualizado 15 de Diciembre de 2010 [Consultado el 10 de Enero de 2012]. Disponible en internet: <<http://perso.wanadoo.es/pictob/comserie.htm>>

TORRES, Leticia Esmeralda. Simulación de Sistemas de Control en JAVA: Bases Conceptuales y Metodológicas para un Laboratorio Virtual. Trabajo de grado Ingeniero de Sistemas. Mérida, Venezuela: Universidad de los Andes, 2006. 107 p. [Consultado el 20 de Marzo de 2011]. Disponible en internet: <http://tesis.ula.ve/pregrado/tde_busca/archivo.php?codArchivo=728>

VELEZ LOPEZ, Juan David y RIOS, Arcesio. Transmisión Serial [en línea]. Medellín, Colombia: Politécnico Colombiano Jaime Isaza Cadavid. [Consultado el 19 de Enero del 2012]. Disponible en internet: <<http://redesvelez.tripod.com/tserial.htm>>

VOEGELE, Jason. Programming Language Comparison [en línea]. [Consultado el 22 de Febrero de 2012]. Disponible en internet: <<http://www.jvoegele.com/software/langcomp.html>>

Wikilearning: comunidades de wikis libres para aprender. Metodologías usadas en ingeniería del software - Paradigma imperativo [en línea]. Actualizado 25 de Octubre de 2007 [Consultado el 16 de Febrero de 2012]. Disponible en internet: <http://www.wikilearning.com/curso_gratis/metodologias_usadas_en_ingenieria_d_el_software/3618-4>

Wikipedia: la enciclopedia libre. Flow Control [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 17 de Septiembre de 2011 [Consultado el 15 de Diciembre del 2011]. Disponible en internet: <http://en.wikipedia.org/wiki/Flow_control>

------. Puerto de red [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 07 de diciembre de 2011 [Consultado el 10 de Enero de 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Puerto_de_red>

------. Puerto Serie [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 14 de noviembre de 2011 [Consultado el 10 de Enero de 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Puerto_serie>

------. Tecnologías de la información y la comunicación [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 09 de enero del 2012 [Consultado el 10 de Enero de 2012]. Disponible en internet:

<http://es.wikipedia.org/wiki/Tecnolog%C3%ADas_de_la_informaci%C3%B3n_y_la_comunicaci%C3%B3n>

WORDPRESS.COM. Mecanismos para la detección de errores [en línea]. Actualizado 11 de Septiembre del 2011 [Consultado el 15 de diciembre del 2011]. Disponible en internet: <<http://mediosdetransmisionyperturbaciones.wordpress.com/4-5/>>

XTREMEPOO. Clases Genéricas (Plantillas) [en línea]. Actualizado 04 de Junio del 2009 [Consultado el 22 de Febrero de 2012]. Disponible en internet: <<http://xtremepoo.blogspot.com/2009/06/clases-genericas-plantillas.html>>

ZARATE REA, Héctor. Paradigmas de Programación [en línea]. Ciudad de México, México: Universidad Nacional Autónoma de México, 2008. 15 p. [Consultado el 21 de Febrero de 2012]. Disponible en internet: <<http://es.scribd.com/doc/9174723/Paradigmas-de-Programacion>>

ANEXOS

Anexo A. Listado de Puertos, entre 49152 y 65535, usados por troyanos

PUERTO	NOMBRE DEL TROYANO
49301	OnLine KeyLogger
50000	SubSARI
50130	Enterprise
50505	Sockets des Troie
50766	Fore, Schwindler
51966	Cafeini
52317	Acid Battery 2000
53001	Remote Windows Shutdown - RWS
54283	SubSeven, SubSeven 2.1 Gold
54320	Back Orifice 2000
54321	Back Orifice 2000 School Bus
55165	File Manager trojan, File Manager trojan, WM Trojan Generator
55166	WM Trojan Generator
57341	NetRaider
58339	Butt Funnel
60000	Deep Throat, Foreplay, Sockets des Troie
60001	Trinity
60008	Lion, T0rn Rootkit
60068	Xzip 6000068
60411	Connection
61348	Bunker-Hill
61466	TeleCommando
61603	Bunker-Hill
63485	Bunker-Hill
64101	Taskman
65000	Devil, Sockets des Troie, Stacheldraht
65390	Eclipse
65421	Jade
65432	The Traitor (= th3tr41t0r)
65432	(UDP) - The Traitor (= th3tr41t0r)
65530	Windows Mite
65534	/sbin/initd

PUERTO	NOMBRE DEL TROYANO
65535	Adore worm, RC1 trojan, Sins

Fuente: Seguridad en la red. Listado de Puertos usados por troyanos [en línea]. [Consultado el 20 de Marzo del 2012]. Disponible en internet: <<http://www.seguridadenlared.org/es/index5esp.html>>

Anexo B. Ejemplos de programación

Paradigma Estructurado. En lo que sigue, se da un ejemplo de un programa que determina si un número es primo en el rango desde 2 hasta n, empleando el algoritmo conocido como La Criba de Eratóstenes implementado en el lenguaje PASCAL, en donde: CONST declara una constante; VAR declara una variable; FOR-DO es un bucle e IF-THEN es un condicional.

```
program Eratosthenes;

const N=1000;

var a:ARRAY[1..N] of boolean;
    i,j,m:word;

begin
    for i:=1 TO N do A[i]:=TRUE;
    m:=trunc(sqrt(N));
    for i:=2 to m do
        if a[i] then for j:=2 to N DIV i do a[i*j]:=FALSE;
        for i:=1 to N do if a[i] then write(i:4);
end.77
```

Paradigma Orientado a Objetos. En seguida se presenta un ejemplo del lenguaje orientado a objetos PYTHON, en donde: la función “input” lee un número digitado por el usuario y “range” determina el rango del bucle for.

```
nums=[ ]
prims=[ ]
num=input(">>> ")
for i in range(2, num+1):
    nums.append(i)
while nums != [ ]:
    n=nums[0]
    prims.append(n)
    n=(float(n))
```

⁷⁷ WIKIPEDIA: la enciclopedia libre. Criba de Eratóstenes [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 04 de Enero del 2012 [Consultado el 11 de Enero de 2012]. Disponible en internet: <http://es.wikipedia.org/wiki/Criba_de_Erat%C3%B3stenes>

```

    for j in nums:
        if ((j % n ) == 0.0):
            nums.remove(j)
print primes78

```

Paradigma Funcional. A continuación se da un ejemplo de un programa que determina si un número es primo usando el lenguaje LISP, en donde: DEFUN es una función que permite crear otras funciones; COND es un condicional con varios predicados; EQ comprueba si dos argumentos son iguales; DIFFERENCE resta, QUOTIENT cociente y REMAINDER es lo demás o el resto.

```

(DEFUN PRIMO(N)
COND((EQ N 2) T) ((EQ N 3) T)
(T (PRIMO1 N (QUOTIENT N 2))))
(DEFUN PRIMO1 (N Y)
COND((EQ (REMAINDER N Y) 0) NUL)
((EQ Y 2) T)
(T (PRIMO1 N (DIFFERENCE Y 1))))

```

Y al ejecutarlo, por ejemplo se tendría:

```

(PRIMO 3) T
(PRIMO 8) NUL79

```

Paradigma Lógico. En el próximo párrafo se muestra un ejemplo de un programa desarrollado en PROLOG, en donde: % es el comienzo de un comentario del usuario; :- es un condicional; <> diferente de, y ?- es una consulta al programa.

```

%%
%% declaraciones
%%

padrede('juan', 'maria'). % juan es padre de maria
padrede('pablo', 'juan'). % pablo es padre de juan

```

⁷⁸ WIKIPEDIA, Op. Cit., Disponible en internet:
http://es.wikipedia.org/wiki/Criba_de_Erat%C3%B3stenes

⁷⁹ WIKILEARNING, Op. Cit., Disponible en internet:
http://www.wikilearning.com/curso_gratis/metodologias_usadas_en_ingenieria_del_software/3618-4

```

padrede('pablo', 'marcela').
padrede('carlos', 'debora').

% A es hijo de B si B es padre de A
hijode(A,B) :- padrede(B,A).

% A es abuelo de B si A es padre de C y C es padre B
abuelode(A,B) :- padrede(A,C), padrede(C, B).

% A y B son hermanos si el padre de A es también el padre
de B y si A y B no son lo mismo
hermanode(A,B) :- padrede(C,A) , padrede(C,B), A <> B.

% A y B son familiares si A es padre de B o A es hijo de B
o A es hermano de B
familiarde(A,B) :- padrede(A,B) ; hijode(A,B) ;
hermanode(A,B).

%%
%% consultas
%%

% juan es hermano de marcela?
?- hermanode('juan', 'marcela').
sí

% carlos es hermano de juan?
?- hermanode('carlos', 'juan').
no

% pablo es abuelo de maria?
?- abuelode('pablo', 'maria').
sí

% maria es abuelo de pablo?
?- abuelode('maria', 'pablo').
No80

```

⁸⁰ BONAPARTE, Ubaldo. Paradigma Lógico [en línea]. San Miguel de Tucumán, Argentina: Universidad Tecnológica Nacional (UTN). Actualizado 02 de Junio del 2006 [Consultado el 16 de Febrero de 2012]. Disponible en internet: <<http://www.frt.utn.edu.ar/sistemas/paradigmas/page34.html>>

Anexo C. Características eléctricas de la señal en el protocolo RS232.

No.	Parámetros	Condiciones	Min	Max
1	Driver, Voltaje de Salida en Circuito Abierto	x	x	25V
2	Driver Apagado, Impedancia de Salida	$-2V \leq V_o \leq 2V$	300 Ω	x
3	Driver, Corriente de Salida en corto-circuito	El conductor será apto para soportar un corto con cualquier otra línea en el cable sin daño a sí mismo o a otro equipamiento.	x	500 mA
4	Driver, Slew Rate*	El tiempo requerido de la señal para pasar de $-3V$ a $+3V$ de la región de transición no podrá exceder 1 ms, o el 4% del tiempo de un bit.	x	30 V/ μ s
5	Capacitancia de carga	Incluyendo la capacitancia del cable. Cuando se está usando un cable normal con una capacitancia de 40 a 50 pF/Pie de longitud, la medida del cable se limita a un máximo de 50 Pies (15 m).	x	2500 pF
6	Receptor, Impedancia de entrada	$3V \leq V_{IN} \leq 25V$	3 K Ω	7 K Ω
7	Driver, Señal MARCA	Respecto a la línea Signal Ground.	-15V	-5V
8	Driver, Señal ESPACIO	Receptor: $3 K\Omega \leq R_L \leq 7 K\Omega$; Voltaje de Circuito Abierto = 0	5V	15V
9	Señal en estado de MARCA (nivel lógico 1)	Respecto a la línea Signal Ground	-15V	-3V
10	Señal en estado de ESPACIO (nivel lógico 0)	Respecto a la línea Signal Ground	+3V	+15V

* Es un efecto no lineal en los amplificadores. Representa la incapacidad de un amplificador para seguir variaciones rápidas de la señal de entrada. También se le define como la velocidad de cambio del voltaje de salida con respecto a las variaciones en el voltaje de entrada. Se expresa típicamente en unidades de V/ μ s.

No.	Parámetros	Condiciones	Min	Max
11	Señal en estado de TRANSICION	Respecto a la línea Signal Ground	-3V	+3V
12	Velocidad de Datos	x	x	20 Kb/s

Fuente: JIMENEZ CHAFLA, Marcela Adriana y ARREGUI JIMENEZ, Gilberth Wilfrido. Diseño e implementación de un sistema de generación de mensajes auditivos para ayudar a personas con discapacidad auditiva y del habla [en línea]. Trabajo de grado Ingeniería en Electrónica y Computación. Riobamba, Ecuador: Escuela Superior Politécnica de Chimborazo, 2009. p. 44-45. [Consultado el 15 de Enero de 2012]. Disponible en internet: <<http://dspace.espace.edu.ec/bitstream/123456789/182/1/38T00170.pdf>>; modificado por el autor.

Obsérvese que los parámetros 7 y 8, permiten 2V de margen de ruido. En la práctica, se utilizan –12V y 12V respectivamente.

Anexo D. Descripción funcional de los pines del DB25 macho

Pin	Nombre	E/S	Descripción
1	GND - Shield Ground (Tierra Protegida)	-	Esta línea está conectada a la tierra de la alimentación del adaptador serial. No debería ser usada como una señal de tierra, y NO DEBE ser conectada a GND. Conecte esta línea a la pantalla del cable (si lo hay). Al conectar esta línea a ambos lados asegura que ningún flujo grande de corriente atraviese GND en caso de un defecto de aislamiento en uno de los lados. ⁸¹
2	TXD - Transmit Data (Datos Transmitidos)	S	La información (caracteres) es transmitida por aquí.
3	RXD - Receive Data (Datos Recibidos)	E	La información (caracteres) es recibida por aquí.
4	RTS - Request to Send (Solicitud para envío)	S	Prepara el DCE para aceptar los datos transmitidos desde el DTE. En aplicaciones half-duplex establece la dirección del canal. Cuando el DCE está listo, activa el CTS.
5	CTS - Clear to Send (Despejado para envío)	E	El DCE informa al DTE que puede comenzar la transmisión. RTS y CTS se usan para moderar el flujo de datos en el DCE.
6	DSR - Data Set Ready (Conjunto de Datos Listo)	E	Indica que el DCE está encendido y listo para funcionar, o no se pueden utilizar en absoluto. Si no se utiliza DSR, debe establecerse a '0' lógico o se programa el DTE para ignorar esta señal.
7	GND – Ground (Tierra)	-	Todas las señales tienen como referencia una tierra común, según lo definido por la tensión en el pin 7. Este conductor puede o no estar conectado a tierra de protección en el interior del DCE.

⁸¹ The Vision and Autonomous Systems Center (VASC). Release 18 [en línea]. Actualizado el 22 de Febrero del 2005 [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://vasc.ri.cmu.edu/old_help/Misc/Serial/serial.txt>

Pin	Nombre	E/S	Descripción
8	DCD – Data Carrier Detect (Detección de Portadora de Datos)	E	Esta señal es relevante cuando el DCE es un módem. Se establece a '0' lógico por el módem cuando la línea telefónica está “descolgada”, una conexión se ha establecido y un tono de respuesta se recibe desde el módem remoto. La señal es un '1' lógico cuando no hay tono de respuesta, o cuando el tono de respuesta es de calidad insuficiente para satisfacer las necesidades del módem local (quizás debido a un canal con ruido).
9	Voltaje Positivo DC para pruebas	-	No se conecta. Reservado para pruebas.
10	Voltaje Negativo DC para pruebas	-	No se conecta. Reservado para pruebas.
11	STF - Select Transmit Channel	S	Escoge cual canal se va a usar para la comunicación, entre el canal primario o el secundario.
12	S.DCD - Secondary Data Carrier Detect	E	Segundo DCD
13	S.CTS - Secondary Clear to Send	E	Segundo CTS
14	S.TXD - Secondary Transmit Data	S	Segundo TXD
15	TCK - Transmitter Clock (Reloj de Transmisión)	E	Esta señal solo es relevante cuando el DCE es un módem y funciona con un protocolo síncrono. El módem genera esta señal de reloj para controlar exactamente la velocidad a la que se transmiten los datos de TxD (pin 2) desde el DTE al DCE. La transición de '1' a '0' lógico en esta línea provoca una transición al elemento de datos siguiente en la línea TxD. El módem genera esta señal de forma continua, excepto cuando se está llevando a cabo las funciones internas de diagnóstico.

Pin	Nombre	E/S	Descripción
16	S.RXD Secondary Receive Data	E	Segundo Reloj de Rx.
17	RCK - Receiver Clock (Reloj de Recepción)	E	Esta señal es similar a TCK (Pin 15), con la excepción de que provee la información temporal para el receptor del DTE.
18	LL - Local Loopback Control (Control de Bucle Local)	S	Esta señal es generada por el DTE y se utiliza para poner el módem en un estado de prueba. Cuando se establece LL a '0' lógico, el módem vuelve a dirigir su señal de salida modulada, que normalmente se introducen en la línea telefónica, de vuelta a sus circuitos de recepción. Esto permite que los datos generados por el DTE sean enviados de vuelta a través del módem local para comprobar el estado de los circuitos del módem. El módem activa TM (pin 25) para reconocer que se ha colocado en condición de local loopback.
19	S.RTS - Secondary Request to Send	S	Segundo RTS.
20	DTR - Data Terminal Ready (Terminal de Datos Listo)	S	Indica que el DTE está encendido y listo para funcionar, o no se pueden utilizar en absoluto. Si lo desea puede ignorar esta señal.
21	RL - Remote Loopback (Bucle Remoto)	S	Esta señal es generada por el DTE y se utiliza para colocar el módem remoto en un estado de prueba. Cuando se establece RL a '0' lógico, el módem remoto redirecciona los datos recibidos de vuelta a su entrada TXD, con lo que remodela los datos recibidos y los devuelve a su fuente. Cuando el DTE inicia como una prueba, los datos transmitidos pasan a través del módem local, la línea telefónica, el módem remoto, y de regreso, para ejercitar el canal y confirmar su integridad. El módem remoto establece el TM (pin 25) del módem local cuando la prueba de RL está en marcha.
22	RI - Ring Indicator (Indicador de llamada)	E	Esta señal es relevante cuando el DCE es un módem, y se establece a un '0' lógico cuando una señal de llamada está siendo recibida de la línea telefónica. El tiempo en que permanece en '0' esta señal será aproximadamente igual a la duración de la señal de llamada, y será '1' lógico entre llamadas o cuando no hay señal de llamada presente.

Pin	Nombre	E/S	Descripción
23	DSR - Data Signal Rate Selector (Selector de Velocidad de Datos)	S	Puede tener su origen tanto en el DTE como en el DCE (pero no ambos), y se utiliza para seleccionar una de las dos velocidades preestablecidas. Si se establece a '0' lógico, selecciona la velocidad más alta de transmisión.
24	XCK - External Transmitter Clock (Reloj de Transmisor Externo)	S	Las señales de reloj son proporcionadas por el dispositivo DTE para el uso de un módem. Esta señal se utiliza sólo cuando TCK y RCK (pines 15 y 17) no están en uso. La lógica de transición de '1' a '0' lógico indica el tiempo-centro del elemento de datos. Las señales de tiempo serán proporcionadas cada vez que el DTE está encendido, independientemente de otras condiciones de señal.
25	TM - Test Mode (Modo de Prueba)	E	Esta señal sólo es relevante cuando el DCE es un módem. Cuando se establece a '0' lógico, indica que el módem está en el modo LL o RL. Otras condiciones internas de autoevaluación también pueden activar TM, y dependen del módem y la red a la que está conectado.

Fuente: CAMI Research Inc. The RS232 Standard: Signal Definitions [en línea]. [Consultado el 15 de Enero del 2012]. Disponible en internet: <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>; modificado por el autor.

Las señales secundarias 12, 13, 14, 16 y 19 son equivalentes a las correspondientes señales en el canal primario de comunicación. La velocidad de transmisión, sin embargo, suele ser mucho más lenta en el canal secundario para una mayor fiabilidad⁸².

⁸² CAMI Research Inc., Op. Cit., Disponible en internet: <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>

Anexo E. Cuadro comparativo de los Robots Manipuladores.

N°	CARACTERÍSTICA TÉCNICA	SCORBOT-ER V PLUS	SCORA-ER 14	MOTOMAN SV3X	LAB-VOLT 5200	REFERENCIA (PROMEDIO)
1	Rango de Movimiento Eje 1	310°	288°	340° (±170°)	315°	313.25°
2	Velocidad Efectiva Eje 1	Sin Información	117°/s	210°/s	Sin Información	163.5°/s
3	Rango de Movimiento Eje 2	165° (+130° / -35°)	218°	195° (+150° / -45°)	155°	183.25°
4	Velocidad Efectiva Eje 2	Sin Información	114°/s	170°/s	Sin Información	142°/s
5	Rango de Movimiento Eje 3	260° (±130°)	182mm (44.1°)*	260° (+190° / -70°)	240°	201°
6	Velocidad Efectiva Eje 3	Sin Información	211mm/s (51.1°/s)**	225°/s	Sin Información	138.05°/s
7	Rango de Movimiento Eje 4	260° (±130°)	±527° (Ilimitado sin Cable Gripper)	360° (±180°)	155°	283.75°
8	Velocidad Efectiva Eje 4	Sin Información	370°/s	300°/s	Sin Información	335°/s
9	Rango de Movimiento Eje 5	Ilimitado (Mecánico) ±570° (Eléctrico)	No existe	270° (±135°)	Ilimitado (±360°)	±400°

* Se tomó 182mm como si fuese la medida de un arco y se tomó como radio el promedio de la longitud del elemento siguiente al eje 3: Scorbot = 220mm, SV3X = 260mm y Lab-Volt = 230mm. $R = (220 + 260 + 230) / 3 = 236.67$; $\theta = S / R = 182\text{mm} / 236.67\text{mm} = 0.769 \text{ rad} = 44.1^\circ$

** Se calcula el tiempo que tarda en desplazarse los 182mm de la siguiente manera: $t = X/V = (182\text{mm})/(211\text{mm/s}) = 0.863\text{s}$. Luego se calcula la velocidad en grados para los 44.1° así: $V = X/t = 44.1^\circ/0.863\text{s} = 51.1^\circ/\text{s}$.

N°	CARACTERÍSTICA TÉCNICA	SCORBOT-ER V PLUS	SCORA-ER 14	MOTOMAN SV3X	LAB-VOLT 5200	REFERENCIA (PROMEDIO)
10	Velocidad Efectiva Eje 5	Sin Información	No existe	300°/s	Sin Información	300°/s
11	Rango de Movimiento Eje 6	No existe	No existe	±350°	No existe	±350°
12	Velocidad Efectiva Eje 6	No existe	No existe	420°/s	No existe	420°/s
13	Velocidad Conjunto	600mm/s	2800mm/s (Horizontal) 330mm/s (Vertical) 2.819,4mm/s*	Sin Información	584mm/s	1.334,5mm/s
14	Área de Trabajo Horizontal Aprox.	967.672mm ²	506.509mm ²	1.092.854mm ²	916.596mm ²	1.062.012mm ²
15	Área de Trabajo Vertical Aprox.	399845mm ²	49140mm ²	533992mm ²	757149mm ²	435031mm ²
16	Apertura del Gripper	75mm (No Almohadillas) 65mm (Si Almohadillas)	64mm	64mm	91mm	73.5mm
17	Max Carga Útil (Con Gripper)	1Kg	2Kg	3Kg	4.5Kg	2.6Kg
18	Repetitividad de Posición	±0.5mm (punta Gripper)	±0.05mm	±0.03mm	±1.25mm	±0.46
19	Temperatura Ambiente de Operación	40°C – 2°C	40°C – 2°C	45°C – 0°C	Sin Información	42°C – 1°C
20	Humedad	Sin Información	Sin Información	80%RH – 20%RH	Sin Información	80%RH – 20%RH
21	Grados de Libertad	5	4	6	5	5

$$* V = \sqrt{330^2 + 2800^2} = \sqrt{7'948.900} = 2.819,4mm/s$$

N°	CARACTERÍSTICA TÉCNICA	SCORBOT-ER V PLUS	SCORA-ER 14	MOTOMAN SV3X	LAB-VOLT 5200	REFERENCIA (PROMEDIO)
22	Programación	ACL – SCORBASE LEVEL 5	ACL	ACL WIN	Level 4 Robotics	ACL
23	Comunicación	RS232	RS232	RS232	RS232	RS232
24	Fuerza Electromotriz	100/110/220VAC	100/120/220/240VAC	200VAC – 230VAC	120VAC	130VAC – 202.5VAC
25	Modo de Uso	Teach Pendant – ACL – SCORBASE LEVEL 5	Teach Pendant – ACL	Teach Pendant – ACL	Teach Pendant – Level 4 Robotics	TP – ACL
26	Media Rango de Movimiento Eje 1 – 3	245°	183.6°	265°	236.67°	232.57°

Anexo F. Comparación de los Lenguajes de Programación

#	Características	Java	C	C#	C++	Objective-C	PHP	Visual Basic	Python	Perl	JavaScript	REFERENCIA
1	Popularidad 2012 (Ranking)	1	2	3	4	5	6	7	8	9	10	5,5
2	Estandarizado ⁸³	SI	SI	SI	SI	NO	NO	NO	SI	NO	SI	SI
3	Paradigma	Imperativo, Orientado a Objetos (Hibrido)	imperativo, procedural	imperativo, funcional, Orientado a Objetos (Hibrido)	imperativo, procedural, Orientado a Objetos (Hibrido)	imperativo, Orientado a Objetos (Hibrido)	imperativo, procedural, Orientado a Objetos (Hibrido)	Imperativo, Soporte Parcial Orientado a Objetos	imperativo, funcional, Orientado a Objetos (Hibrido)	imperativo, funcional, procedural, Orientado a Objetos (Hibrido)	imperativo, funcional	Orientado a Objetos (Hibrido)
4	Tipado Estático / Dinámico	Estático	Estático	Estático	Estático	Estático	Dinámico	Estático	Dinámico	Dinámico	Dinámico	Estático
5	Clases Genéricas	No	No	Si	Si	Si	Si*	No	Si	Si	Si	Si
6	Herencia	Clase única, Múltiples interfaces	No	Clase única, Múltiples interfaces	Múltiple	Clase única, Múltiples Interfaces	Clase única, Múltiples Interfaces	Ninguna	Múltiple	Múltiple	basada en prototipos	Clase única, Múltiples interfaces

⁸³ Wikipedia: la encyclopedía libre. Comparison of programming languages [en línea]. Florida: Wikimedia Foundation, 2006. Actualizado 16 de Febrero del 2012 [Consultado el 24 de Febrero del 2012]. Disponible en internet: <http://en.wikipedia.org/wiki/Comparison_of_programming_languages>

* Los programas de escritura dinámica soportan la programación genérica inherentemente.

#	Características	Java	C	C#	C++	Objective-C	PHP	Visual Basic	Python	Perl	JavaScript	REFERENCIA
7	Recolección de basura	Marque y barra o Generacional	Ninguna	Marque y barra o Generacional	Ninguna	Conteo de referencia, Marque y barra	Conteo de referencia	Conteo de referencia	Conteo de referencia	Conteo de referencia	Marque y barra	Conteo de referencia
8	Control de Acceso	public, protected, "package", private	Local, Global, Parámetro Formal	public, protected, private, internal, protected internal	public, protected, private, "friends"	public, protected, private	public, protected, private	public, private	Name Mangling	Ninguna	Ninguna	public, protected, private
9	Múlti-hilo	Si	Librerías	Si	Librerías	Mediante NSOperation	No	No	Si	No	Librerías	Si
10	Integración de Lenguaje	C, varios C++	No	Todos los Lenguajes .NET	C, Assembler	C	MySQL, PostgreSQL	C (vía DCOM)	C, C++, Java	C, C++	No	C
11	Seguridad Incorporada	Si	No	Si	No	No	Si	No	No	Si (perlsec)	No	No
12	Nivel de Lenguaje Capers Jones ⁸⁴	6	2.5	6	6	12	-	11	15	15	-	9

⁸⁴ SCHENONE, Marcelo Hernán. Diseño de una Metodología Ágil de Desarrollo de Software [en línea]. Trabajo de grado Ingeniería en Informática. Buenos Aires, Argentina: Universidad de Buenos Aires, 2004. p. 171-174. [Consultado el 24 de Febrero de 2012]. Disponible en internet: <<http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>>

#	Características	Java	C	C#	C++	Objective-C	PHP	Visual Basic	Python	Perl	JavaScript	REFERENCIA
13	Licencia	GNU GPL (Software Libre)	(Software Libre)	(Software Libre)	(Software Libre)	GNU GPL (Software Libre)	PHP y GNU GPL (Software Libre)	Privativo	PSFL (Software Libre)	Licencia Artística y GNU GPL (Software Libre)	Netscape Public License y GNU GPL (Software Libre)	GNU GPL (Software Libre)

Fuente: VOEGELE, Jason. Programming Language Comparison [en línea]. [Consultado el 22 de Febrero de 2012]. Disponible en internet: <<http://www.jvoegele.com/software/langcomp.html>>; modificado por el autor.

Anexo G. Nivel Carpers Jones

Así como existen medidas de calidad que comparan los diferentes productos que hay en el mercado, también las hay para el *software*. Dos de estas métricas son: los Puntos Funcionales (del inglés *Function Points* - FP) de Albrecht y los Puntos de Características (del inglés *Feature Points* - FP) de Jones. Esta última se tiene en cuenta en este proyecto para seleccionar el software a usar. Para comenzar se hace la diferencia entre los dos FP, definiéndolos a lo largo de la presente sección. Esto es importante para entender mejor como mide Jones los niveles de *software* incluidos en el cuadro 1, la cual, se considera un fragmento de la versión 7 que es producida por el SPR (*Software Productivity Research* – Investigación en Productividad de Software).

Cuadro 45. Puntos de Característica para diferentes lenguajes de programación

LENGUAJE	NIVEL	DECLARACIONES PROMEDIO POR PUNTOS DE FUNCION
1st Generación por defecto	1.00	320
2nd Generación por defecto	3.00	107
3rd Generación por defecto	4.00	80
4th Generación por defecto	16.00	20
5th Generación por defecto	70.00	5
Lenguaje máquina	0.50	640
Java	6.00	53
C #	6.00	53
Objective-C	12.00	27
Visual Basic 5	11.00	29
PERL	15.00	21
C	2.50	128
C++	6.00	53
PHP	-	-
Python	15.00	21
JavaScript	-	-

Fuente: HUALING, Li. *Programming Languages Table—By Capers Jones* [en línea]. Actualizado el 05 de Septiembre del 2006 [Consultado el 13 de Marzo del 2012]. Disponible en internet: <http://blog.csdn.net/li_hualing/article/details/1179557>; modificado por el autor.

Hay que tener en cuenta que, los lenguajes de programación están en constante evolución, es por ello, que los niveles suben y menos declaraciones se requieren para codificar un Punto de Función.

Puntos Funcionales. Las primeras investigaciones acerca de las medidas del software fueron realizadas en la década de los setenta en la empresa IBM (del inglés international business machine) por Allan J. Albrecht. Hacia finales del año 1979, Albrecht descubrió una técnica que permite medir el tamaño de una aplicación independiente de la tecnología usada, del lenguaje de programación, de la metodología de desarrollo y de la capacidad del equipo de trabajo para desarrollar la aplicación. Es así que la denominó FPA (del inglés *function point analysis*). Fue la primera técnica capaz de medir funcionalidad en el software y proporcionar valores aproximados del tamaño en etapas tempranas del desarrollo. Esta técnica se basa en la descomposición de un sistema en componentes más pequeños, de tal forma que estos puedan ser medidos y/o analizados con mayor facilidad. El método intenta enfocar al software desde el punto de vista del usuario; y la medición, se realiza cuantificando la funcionalidad brindada a dicho usuario, partiendo fundamentalmente de diseños lógicos. En el marco del cálculo de los FP, las aplicaciones son divididas en cinco componentes⁸⁵:

Salidas Externas (del inglés *External Output* - EO). Una EO es un proceso elemental que envía datos o información de control fuera de los límites de la aplicación. Su objetivo es presentar información al usuario luego del procesamiento lógico de datos o de la información de control obtenida. El procesamiento lógico debe contener al menos una fórmula o cálculo matemático o crear datos derivados de los obtenidos; además una EO podría mantener por lo menos un ILF o alterar el comportamiento del sistema⁸⁶. Las reglas para identificar EOs, se muestran en el cuadro 2.

⁸⁵ OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 22, 23 y 27. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

⁸⁶ Ibíd., p. 30-31. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Cuadro 46. Reglas para identificar EOs

Regla	Como Identificar EOs
R1	Los datos se envían a través de un proceso elemental fuera de los límites de la aplicación.
R2	Los datos mantienen un ILF a través de un proceso elemental de la aplicación.
R3	Un proceso es la unidad más pequeña de actividad significativa para el usuario final.
R4	El proceso identificado debe observar algunas de las siguientes reglas: ✓ Su lógica de proceso es única con respecto a otras EOs de la aplicación. ✓ Los elementos de datos identificados son distintos a los de otras EOs de la aplicación. ✓ Los ILF referenciados son diferentes.
R5	Deben cumplirse al menos una de las siguientes condiciones: ✓ El proceso elemental contiene al menos una fórmula matemática. ✓ El proceso crea datos derivados. ✓ El proceso que genera la salida mantiene al menos un ILF. ✓ El proceso que genera la salida altera el comportamiento del sistema.
R6	La transferencia de datos a otras aplicaciones se cuenta como EO.
R7	Los informes escritos y en línea se cuentan como salidas independientes.
R8	Los gráficos se cuentan como salidas independientes.

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 31. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Consultas Externas (del inglés *External Queries* - EQ). Las EQ identifican a procesos elementales o acciones que envían datos fuera de los límites de la aplicación que se quiere medir. Ningún ILF es mantenido mientras se procesa una acción de una EQ ni tampoco se altera el comportamiento del sistema. Su objetivo es presentar información al usuario mediante una simple obtención de datos o información de control. La diferencia con EOs radica fundamentalmente en que su procesamiento lógico no debe contener fórmulas o cálculos matemáticos, ni

tampoco debe crear datos derivados de los obtenidos⁸⁷. Las reglas que se aplican para identificar EQs, se muestran en el cuadro 3.

Cuadro 47. Reglas para identificar EQs

Regla	Como Identificar EQs
R1	Una petición entra en los límites de la aplicación.
R2	El resultado de esa petición sale de los límites de la aplicación.
R3	Existe recuperación de datos.
R4	Los datos recuperados no contienen datos derivados.
R5	El proceso lógico no contiene cálculo o fórmulas matemáticas.
R6	El proceso que genera la EQs no mantiene ningún ILF y no altera el comportamiento del sistema.
R7	La petición de entrada y el resultado de salida juntos, hacen del proceso la unidad de actividad más pequeña que es significativa para el negocio del usuario.
R8	El proceso es autocontenido y deja a la aplicación en estado consistente.
R9	El proceso no actualiza ILFs.
R10	El proceso debe verificar las siguientes reglas: <ul style="list-style-type: none"> ✓ La lógica del proceso sobre la entrada y la salida es única con respecto a otras EQs. ✓ Los elementos de datos que forman la entrada y la salida son distinto respecto a los de otras consultas de la aplicación. ✓ Los ficheros lógicos referenciados son distintos.

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 32. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

⁸⁷ Ibíd., p. 31. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Entradas Externas (del inglés *External Inputs – EI*).** Una EI es un proceso elemental o una acción que procesa datos o información de control que provienen desde afuera de los límites de la aplicación a medir. Su procedencia puede ser vía un programa externo (se ejecuta fuera de los límites) o mediante la acción de ingreso de datos de un usuario a través de un dispositivo de entrada (por ejemplo teclado, ratón, o cualquier dispositivo de entrada de datos). Las entradas externas para que sean consideradas como tales deben mantener por lo menos un ILF y/o alterar el comportamiento del sistema⁸⁸. Las reglas que se aplican para identificar las EIs se muestran en el cuadro 4.

Cuadro 48. Reglas para identificar EIs

Regla	Como Identificar EIs
R1	Los datos se reciben desde afuera de los límites de la aplicación
R2	Los datos mantienen un ILF a través de un proceso elemental de la aplicación
R3	Un proceso es la unidad más pequeña de actividad significativa para el usuario final.
R4	El proceso identificado debe observar algunas de las siguientes reglas: <ul style="list-style-type: none"> ✓ Su lógica de proceso es única con respecto a otras EIs de la aplicación. ✓ Los elementos de datos identificados son distintos a los de otras EIs de la aplicación. ✓ Los ILF referenciados son diferentes.

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 30. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Archivos Lógicos Internos (del inglés *Internal Logical File – ILF*).** Los ILF representan a grupos de datos relacionados lógicamente o información de

⁸⁸ Ibíd., p. 30. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

control que se mantiene dentro de los límites de la aplicación⁸⁹. Las reglas para identificar a ILFs se muestran en el cuadro 5.

Cuadro 49. Reglas para identificar ILFs

Regla	Como Identificar ILFs
R1	Grupo de datos lógicos identificables por el usuario que cubre de manera completa requisitos específicos de este.
R2	Grupo de datos lógicos mantenidos dentro de los límites de la aplicación.
R3	Grupo de datos lógicos mantenido o modificado mediante un proceso elemental de la aplicación.
R4	Grupo de datos lógicos que no se ha contabilizado como ELFs.

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 28. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Archivos Lógicos Externos (del inglés *External Logical File* – ELF).** Los ELFs representan a grupos de datos relacionados lógicamente o información de control reconocida por el usuario que puede ser referenciada y mantenida fuera de los límites de la aplicación. Este último concepto representa la principal diferencia entre un ILF y un ELF⁹⁰. Las reglas para identificar a ELFs, se muestran en el cuadro 6.

Cuadro 50. Reglas para identificar ELFs

Regla	Como Identificar ELFs
R1	Grupo de datos lógicos identificables por el usuario que cubre de manera completa sus requisitos.

⁸⁹ Ibíd., p. 28. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

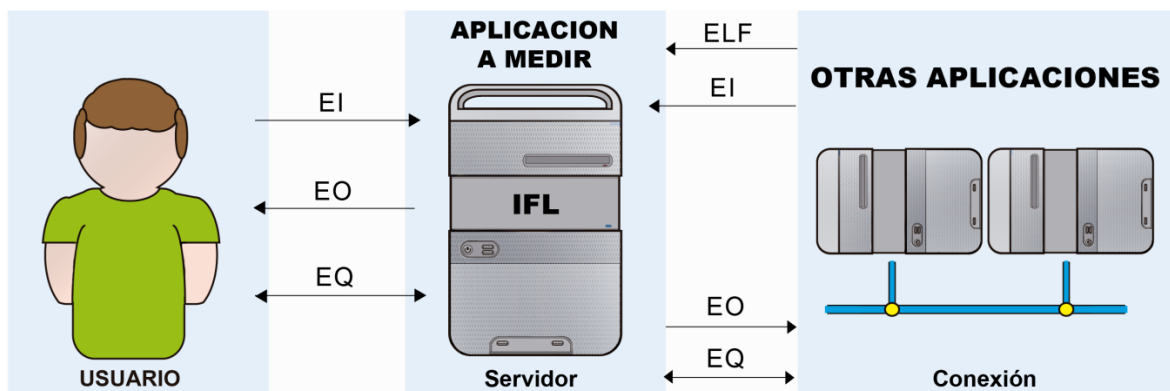
⁹⁰ Ibíd., p. 28. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Regla	Como Identificar ELFs
R2	Grupo de datos lógicos referenciados pero externos a la aplicación que está siendo contada.
R3	Grupo de datos lógicos mantenidos fuera de los límites de la aplicación.
R4	Grupo de datos lógicos que no ha sido contado como un ILF.

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 29. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Los tres primeros de esta lista (EO, EQ y EI) corresponden al grupo denominado “funciones transaccionales” que actúan sobre archivos (grupos lógicos de datos) actualizando la información contenida en estos. Los dos últimos componentes (ILF y ELF) son las entidades, almacenes o “grupos lógicos de datos” (comúnmente denominados archivos). En la figura 1 se muestra la funcionalidad desde el punto de vista del usuario⁹¹.

Figura 42. Funcionalidad desde el punto de vista del usuario



Fuente: ZORRILLA PRIETO, Julián Andrés.

⁹¹ Ibíd., p. 28. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Muchas veces se relacionan a los ILFs y a los ELFs con archivos físicos. Esto es un grave error, ya que cuando se hace referencia a entidades puede no haber una relación directa con aquellos. Por ejemplo, si existe una entidad persona, los atributos y valores de esta entidad pueden estar diseminados en una o varias tablas o archivos físicos de datos (por ejemplo, personas, tipos de personas, domicilios de personas, etc.). En estos casos se debe contar como un solo ILF o ELF y no tres o más⁹².

- **Determinación de la complejidad para grupos de Datos lógicos.** La complejidad de los grupos de datos lógicos se sustenta sobre dos parámetros: tipo de dato elemental (o primario) o DET (del inglés *Data Element Type*) y tipo de registro elemental (o primario) o RET (del inglés *Record Element Type*).

Un DET es un atributo (o campo lógico) reconocible por el usuario que compone una entidad. Para el conteo de DETs, se aplican las siguientes reglas:

- Las claves foráneas para referenciar otro ILF según los requerimientos también se cuenta como DET.
- Un atributo (o campo lógico) único pero almacenado en más de un archivo físico o tabla, se cuenta una sola vez.
- Ídem para los que se repiten.
- Los atributos (o campos lógicos) que aparecen por la aplicación de técnicas de implementación pero que no son reconocibles por el usuario no se cuentan como DETs.

Un RET representa a un subgrupo de datos elementales reconocibles por el usuario. Se debe contar como un RET por cada uno de estos. Existen dos tipos de subgrupos:

⁹² Ibíd., p. 29. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Opcional.** Son aquellos que el usuario tiene la opción de usar solamente uno o ningún subgrupo en un proceso elemental que agrega o crea una instancia de datos.
- **Obligatorios.** Son subgrupos que el usuario debe usar al menos uno.
Las reglas que se aplican en el momento de contar RETs son las siguientes:
- Contar como un RET a cada subgrupo opcional u obligatorio de un ILF y, o ELF.
- Si no hay subgrupos se debe contar a un ILF o ELF como un RET.
- Una vez que se han identificado los DETs y RETs, de cada archivo lógico, se debe determinar su complejidad⁹³ de acuerdo a los valores del cuadro 7.

Cuadro 51. Determinación de la complejidad de los ILFs y ELFs

	1 a 19 DETs	20 a 50 DETs	51 o más DETs
1 RET	Baja	Baja	Media
2 a 5 RETs	Baja	Media	Alta
6 o más RETs	Media	Alta	Alta

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 30. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Determinación de la complejidad para Funciones Transaccionales.** La determinación de la complejidad de las Els, se realiza considerando a dos parámetros. Uno de ellos son los DETs o tipos de datos elementales (visto en párrafos anteriores) y el otro corresponde a los tipos de archivos referenciados

⁹³ Ibíd., p. 29-30. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

o FTR (del inglés *File Type Referenced*). Las reglas que se deben aplicar para el conteo de DETs, son las siguientes:

- Es considerado como DET todo campo reconocible por el usuario y no repetido que cruza las fronteras de la aplicación.
- Una definición del comienzo de la ejecución del proceso (si existen varias formas contar solamente una).
- Mensajes de error que se muestran durante la ejecución del proceso (si hay varios mensajes de error contar solamente uno).

Con respecto a los FTRs se deben contar solo uno por cada archivo interno leído y, o mantenido por el proceso de entrada⁹⁴. El cuadro 8 muestra el grado de complejidad de las Els en función de la cantidad de DETs y FTRs.

Cuadro 52. Determinación de la complejidad de Els en función de DETs y FTRs

	1 a 4 DETs	5 a 15 DETs	16 o más DETs
1 FTR	Baja	Baja	Media
2 FTR	Baja	Media	Alta
3 o más FTRs	Media	Alta	Alta

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 33. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Para EOs y EQs que cruzan los límites de la aplicación, se deben aplicar las siguientes reglas:

⁹⁴ Ibíd., p. 32. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- Contar un DET por cada campo reconocible por el usuario no repetido, que ingresa a la aplicación y que se requiere para especificar cuándo, que o como deben ser recuperados los datos de salida.
- Contar un DET por cada campo reconocible por el usuario no repetido que sale de los límites de la aplicación.
- Un dato que entra y sale de la aplicación debe ser contado solamente una vez.
- Contar un DET por todos los mensajes de error del proceso.
- Contar un DET por la iniciación del proceso, aun habiendo varias formas de hacerlo.
- No se deben contar datos recuperados o derivados por el sistema que no cruzan los límites de la aplicación.
- Contar un FTR por cada archivo interno leído por el proceso de salida o consulta⁹⁵.

En el cuadro 9 se muestra la cantidad de DETs y FTRs identificados y relacionados con su nivel de complejidad.

Cuadro 53. Determinación de la complejidad de EOs y EQs en función de DETs y FTRs

	1 a 5 DETs	6 a 19 DETs	20 o más DETs
1 FTR	Baja	Baja	Media
2 a 3 FTRs	Baja	Media	Alta
4 o más FTRs	Media	Alta	Alta

⁹⁵ Ibíd., p. 33. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 33. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Aplicación de fórmulas para conteo de los Puntos de Función sin ajustar.**
Una vez identificados los componentes, su cantidad y su nivel de complejidad (bajo, medio o alto), se realiza la cuenta de los FP sin ajustar⁹⁶. En el cuadro 10 se muestran los valores (o pesos) que puede adquirir cada uno de los componentes en función de su complejidad.

Cuadro 54. Valores de los pesos para FP sin ajustar

Componente	Complejidad	Peso	Cantidad	Subtotal (peso x cantidad)
ILF	Alta	15	---	---
	Media	10	---	---
	Baja	7	---	---
ELF	Alta	10	---	---
	Media	7	---	---
	Baja	5	---	---
EI	Alta	6	---	---
	Media	4	---	---
	Baja	3	---	---
EO	Alta	7	---	---
	Media	5	---	---
	Baja	4	---	---
EQ	Alta	6	---	---
	Media	4	---	---
	Baja	3	---	---
Total de FP sin ajustar es la sumatoria del producto de los pesos, por las cantidades según sea el componente.				Total de puntos de FP sin ajustar

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas

⁹⁶ Ibíd., p. 33. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 34. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Calibración de los FP en base a las características generales de la aplicación.** Los FP de Albrecht, sin ajustar deben ser calibrados (o ajustados) con otros 14 componentes, cuyos valores dependen del entorno de desarrollo. Estas características en su conjunto se denominan también factores de ajuste (del inglés *factors adjust* – FA).

Estos factores de ajuste pueden tomar valores entre cero (0) y cinco (5). Es decir, cero cuando la característica del componente está ausente o es nula y cinco cuando puede alcanzar una valoración máxima; también se pueden asociar porcentajes⁹⁷, como se muestra en el cuadro 11.

Cuadro 55. Factores de influencia en el sistema para cálculo de FA

VALOR DEL FA	INFLUENCIA EN EL SISTEMA	PORCENTAJE QUE AFECTA O ES REQUERIDO POR LA APLICACIÓN
0	Ninguna	0%
1	Insignificante	1% - 20%
2	Moderada	21% - 40%
3	Media	41% - 60%
4	Significativa	61% - 80%
5	Fuerte	81% - 100%

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 35. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

A continuación se enuncian y se realiza una breve descripción de cada una de las 14 características y los posibles valores que debería considerar el encargado de un proyecto a la hora de realizar la estimación del tamaño del software.

⁹⁷ Ibid., p. 34-35. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Comunicación de datos.** Son los datos o información de control que la aplicación utiliza, se envía o recibe a través de las facilidades (sistema) de comunicación. A modo de ejemplo se puede citar a aquellas aplicaciones que se desarrollan para entidades bancarias y que registran transacciones en distintos puntos geográficos, las cuales hacen uso en forma masiva de las comunicaciones de datos (ya sea vía satélite o vía telefónica). Los valores que puede tomar la característica se imprimen en el cuadro 12.

Cuadro 56. Valores que puede tomar la característica Comunicación de Datos

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
La aplicación no usa conexiones remotas	0
Impresión o entrada de datos remota.	1
Teleproceso interactivo.	2-3
Teleproceso interfaz a un proceso batch.	4
La aplicación es interactiva y en forma remota predominantemente.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 35. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Función distribuida.** Una aplicación que se ejecuta sobre un solo procesador, se puede considerar monolítica; pero si se ejecuta en distintos procesadores y persigue un único objetivo, significa que los componentes (o los datos) de la aplicación están distribuidos. Un ejemplo de este tipo de requerimiento es un motor de búsqueda Web donde el procesamiento está repartido en más de un procesador⁹⁸. Los valores que puede tomar la característica se muestran en el cuadro 13.

⁹⁸ Ibíd., p. 35. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Cuadro 57. Valores que puede tomar la característica Función Distribuida

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
La aplicación no ayuda a la transferencia de datos o a la función de procesamiento entre los componentes del sistema.	0
La aplicación prepara datos para el usuario final de otro procesador.	1
Los datos se preparan para transferencia; se transfieren, y se procesan en otro componente del sistema.	2-4
Las funciones de procesamiento se realizan dinámicamente en el componente más apropiado del sistema.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 36. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Rendimiento.** Está referido a la importancia del tiempo de respuesta dentro de todo el sistema. El tiempo de respuesta se verá influenciado por el diseño, desarrollo, instalación y soporte de la aplicación⁹⁹. Los valores que puede tomar la característica se registran en el cuadro 14.

Cuadro 58. Valores que puede tomar la característica Rendimiento

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
Análisis y diseño de las consideraciones del rendimiento son estándar. No se precisan requerimientos especiales por parte del usuario.	0 – 3
En la fase de diseño se incluyen tareas del análisis del rendimiento para cumplir los requerimientos del usuario.	4
Además de los dos ítems anteriores, se utilizan herramientas de análisis del rendimiento en el diseño, desarrollo e instalación.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software.

⁹⁹ Ibíd., p. 36. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 36. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Configuración utilizada masivamente.** Se refiere a la importancia del entorno y del uso del sistema. Por ejemplo, un sistema de ventas de un supermercado con líneas de caja conectadas a un servidor, demanda una alta tasa de ocupación de recursos (memoria, disco, procesador, comunicaciones, etc.), lo cual provoca restricciones en el hardware¹⁰⁰. Los valores que puede tomar la característica se enuncian en el cuadro 15.

Cuadro 59. Valores que puede tomar la característica Configuración utilizada masivamente

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
La aplicación corre en una máquina estándar sin restricciones de operación.	0 – 3
Restricciones de operación requieren características específicas de la aplicación en el procesador central.	4
Además hay restricciones específicas a la aplicación en los componentes distribuidos del sistema.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 36. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Tasas de transacción.** Una alta llegada de transacciones provoca problemas más allá de los de la característica tres (3). Como ejemplo de este tipo de características están los sistemas bancarios donde a determinadas horas (denominadas pico) las transacciones tienen un volumen considerable¹⁰¹. Los valores que puede tomar la característica se presentan en el cuadro 16.

¹⁰⁰ Ibíd., p. 36. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

¹⁰¹ Ibíd., p. 36-37. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Cuadro 60. Valores que puede tomar la característica Tasas de transacción

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
Las tasas son tales que las consideraciones de análisis de rendimiento son estándares.	0 – 3
En la fase de diseño se incluyen tareas de análisis de rendimiento para verificar las altas tasas de transacciones.	4
Además se utilizan herramientas de análisis de rendimiento.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 37. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Entrada en línea de datos.** Se refiere al tipo de entradas (interactivas). Por ejemplo un programa de aceptación de solicitudes de préstamos donde el usuario introduce los datos del solicitante y debe obtener una respuesta inmediata acerca de la aceptación o no de los perfiles exigidos en el trámite¹⁰². Los valores que puede tomar la característica se imprimen en el cuadro 17.

Cuadro 61. Valores que puede tomar la característica Entrada en línea de datos

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
Hasta el 15% de las transacciones tienen entrada interactiva.	0 – 2
15% al 30% de las transacciones tienen entrada interactiva.	3 – 4
30% al 50% de las transacciones tienen entrada interactiva.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 37. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

¹⁰² Ibid., p. 37. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Diseño para la eficiencia de usuario final.** Determina si las entradas interactivas de datos requieren que se lleven a cabo sobre múltiples pantallas, u operaciones especiales. Ejemplo de este tipo de transacciones son las que se llevan a cabo en pantallas sensibles al tacto. Existen operaciones, especialmente aquellas relacionadas con la venta de pasajes en línea, pago con tarjeta de crédito, operaciones en cajeros automáticos, etc., donde el usuario debe interactuar con múltiples pantallas¹⁰³. Los valores que puede tomar la característica se relacionan en el cuadro 18.

Cuadro 62. Valores que puede tomar la característica Diseño para la eficiencia de usuario final

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
No se especifican requerimientos especiales	0 – 3
Se incluyen tareas de diseño para la consideración de factores humanos	4
Además se utilizan herramientas especiales o de prototipado para promover la eficiencia	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 37. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Actualización en línea.** Se refiere a la necesidad de la aplicación de actualizar los datos en línea. Se da muchos de estos casos en los sistemas de reservas de pasajes, para evitar vender dos veces la misma plaza¹⁰⁴. Los valores que puede tomar la característica se describen en el cuadro 19.

¹⁰³ Ibid., p. 37. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

¹⁰⁴ Ibid., p. 37. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Cuadro 63. Valores que puede tomar la característica Actualización en línea

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
No existe actualización en línea.	0
Actualización en línea de los archivos de control. El volumen de actualización es bajo y la recuperación fácil.	1 – 2
Actualización en línea de la mayoría de los archivos internos lógicos.	3
Además es esencial la protección contra la pérdida de datos.	4
Además se considera el coste de recuperación de volúmenes elevados de grupos de datos.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 38. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Complejidad del procesamiento.** Se refiere a una cantidad considerable de decisiones lógicas que debe tomar la aplicación, o realizar muchos cálculos matemáticos o manejar un número grande de excepciones. Las características identificables de la aplicación son:
 - Mucho procesamiento matemático y/o lógico.
 - Procesamiento complejo de las entradas.
 - Procesamiento complejo de las salidas.
 - Muchas excepciones de procesamiento, muchas transacciones incompletas y mucho reprocesamiento de las transacciones.
 - Procesamiento de seguridad y/o control sensitivo.

De acuerdo a las características mencionadas en el párrafo anterior la variable Complejidad de procesamiento puede tomar los valores que están en¹⁰⁵ el cuadro 20.

Cuadro 64. Valores que puede tomar la característica Complejidad de Procesamiento

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
No se aplica ninguna característica.	0
Se aplica alguna característica.	1
Se aplican dos características.	2
Se aplican tres características.	3
Se aplican cuatro características.	4
Se aplican todas.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 38. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Utilizable en otras aplicaciones.** El código se diseña para que sea compartido o utilizable por otras aplicaciones. Por ejemplo una función de altas, bajas y modificaciones (ABM) que la mayoría de las aplicaciones de nóminas de personas la usa¹⁰⁶. Los valores que puede tomar la característica son adicionados en el cuadro 21.

Cuadro 65. Valores que puede tomar la característica Utilizable en otras aplicaciones

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
Una aplicación local que responde a las necesidades de una organización usuaria.	0 – 1

¹⁰⁵ Ibid., p. 38. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

¹⁰⁶ Ibid., p. 38. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
La aplicación utiliza o produce módulos comunes que consideran más necesidades que las del usuario.	2 – 3
Además, la aplicación se "empaquetó" y documentó con el propósito de fácil reutilización.	4 – 5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 39. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Facilidad de Instalación.** Cuando se necesita de un desarrollo adicional para la instalación de la aplicación, se le asignará máxima puntuación¹⁰⁷. Los valores que puede tomar la característica están en el cuadro 22.

Cuadro 66. Valores que puede tomar la característica Facilidad de Instalación

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
No se requieren por parte del usuario facilidades especiales de conversión e instalación.	0 – 1
Los requerimientos de conversión e instalación son descritos por el usuario y se proporcionan guías de conversión e instalación.	2 – 3
Además se proporcionan y prueban herramientas de conversión e instalación.	4 – 5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 39. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Facilidad de operación.** Los valores que puede tomar la característica están descritos en el cuadro 23.

¹⁰⁷ Ibid., p. 39. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Cuadro 67. Valores que puede tomar la característica Facilidad de operación

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
No se especifican por parte del usuario consideraciones específicas de operación.	0
Se requieren, proporcionan y prueban procesos específicos de arranque, back-up y recuperación.	1 – 2
Además la aplicación minimiza la necesidad de actividades manuales, tales como instalación de cintas y papel	3 – 4
La aplicación se diseña para operación sin atención.	5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 39. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Puestos múltiples.** Cuando el sistema se construye para ser ejecutado en diferentes regiones, con diferentes monedas o idiomas, aumenta su puntuación, ya que esta facilidad le añade un cierto grado de complejidad¹⁰⁸. Los valores que puede tomar la característica se añaden en el cuadro 24.

Cuadro 68. Valores que puede tomar la característica Puestos múltiples

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
El usuario no requiere la consideración de más de un tipo de puesto.	0
Se incluyeron necesidades de varios tipos de puestos en el diseño.	1 – 3
Se proporciona documentación y plan de apoyo para soportar la aplicación en varios tipos de puestos.	4 – 5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 39.

¹⁰⁸ Ibid., p. 39. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

[Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

- **Facilidad de cambio.** Cuando se debe desarrollar la aplicación para que se adapte a los cambios y le facilite la operación al usuario, alcanza máximos puntajes¹⁰⁹. Los valores que toma la característica se pueden consultar en el cuadro 25.

Cuadro 69. Valores que puede tomar la característica Facilidad de cambio

DESCRIPCIÓN DE LA CARACTERÍSTICA	VALOR
No hay requerimientos especiales del usuario para minimizar o facilitar el cambio.	0
Se proporciona capacidad de consulta flexible.	1 – 3
Datos importantes de control se mantienen en tablas que son actualizadas por el usuario a través de procesos en línea interactivos.	4 – 5

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 40. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Una vez obtenido los valores para las 14 características, de acuerdo al entorno de desarrollo del software, se calcula el FA para la aplicación que está siendo medida. Dicho cálculo se realiza siguiendo los procedimientos de la ecuación 1.

Ecuación 16. Cálculo de los FPA

$$FA = 0.65 + 0.01 \left(\sum_{i=1}^{14} Caracteristicas_i \right)$$

¹⁰⁹ Ibid., p. 39. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

$$FPA = (FP_{\text{sinajustar}})FA$$

Dónde:

FA = factor de ajuste.

FP = puntos de función sin ajustar.

FPA = puntos de función ajustados o totales.

Los valores 0.65 y 0.01 no son arbitrarios; son el resultado de cientos de mediciones realizadas en diferentes proyectos. El FA de los FPA, puede tener una influencia que va desde 0.65 hasta 1.35; es decir si todas las características toman valor nulo (cero) los FP se multiplican por 0.65; es decir, reciben desde la calibración una influencia negativa. Sin embargo si toman el máximo valor (cinco para cada una) se multiplican los FP por 1.35; es decir, reciben desde la calibración una influencia positiva.

El software que mejor se adapta a este tipo de mediciones es el software de gestión, que procesa información comercial. Si bien es cierto, los puntos de función de Albrecht, no son aplicables a todos los tipos de software, la mayoría de los métodos que incorporan facilidades para distintas aplicaciones se basan en la técnica de Albrecht, modificando ligeramente algunos aspectos o incorporando nuevos elementos a los mencionados.

Cada punto de funcionalidad (PF) representa a una cantidad de líneas de código, las que en definitiva determinan el tamaño de la aplicación a desarrollar. En las etapas tempranas del desarrollo, esa medida es aproximada y sirve, entre otras cosas, para decidir la realización o no del proyecto. Los PF no solamente permiten obtener el tamaño del producto software, sino que además se pueden hacer estimaciones del costo, esfuerzo y duración aproximados¹¹⁰.

¹¹⁰ Ibid., p. 40. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Puntos de característica (*Feature Points*). Las investigaciones de los puntos de características, comenzaron aproximadamente en el año 1999. Utilizaba la lógica de los FP, para estimar el tamaño de *software* y se aplicó en un principio a sistemas operativos, sistemas de control de telefonía y aquellos sistemas denominados de tiempo real. Para evitar confusiones con el método de Albrecht, a este método se lo denominó puntos de características (*FP* del inglés *feature points*). Desde un principio, el método arrojó resultados favorables en los experimentos realizados en *software* embebido, *software* de tiempo real, sistemas CAD, y algunas aplicaciones de *software* de gestión. Precisamente, con el crecimiento de la industria del *software* y con su correspondiente masificación, comenzaron a surgir requerimientos que no eran necesariamente de gestión administrativa o comercial (por nombrar los más conocidos). Por consiguiente, hubo que desarrollar *software* diferente; el cual debía ser medido y/o estimado.

Algunos ejemplos de aplicaciones en las que el método de Albrecht no era del todo exacto, y que permitió el nacimiento de esta nueva técnica entre otras son:

- Software de tiempo real para control de armamento en las fuerzas armadas.
- Nuevos sistemas operativos.
- Software embebido.
- Software de comunicaciones (por ejemplo para control de telefonía).
- Software de control de dispositivos.

Cuando se aplica los FP de Albrecht a este tipo de desarrollos, no se obtiene una estimación del todo exacta. La razón fundamental es que aquellos fueron desarrollados para sistemas de *software* de gestión; y con otro tipo de *software*, la cuenta de los FP no es del todo confiable. La técnica de Jones, puede sustituir a los FP de Albrecht, agregando algunos parámetros de tal forma que permita representar a aquellos algoritmos complejos o procesos que requieren de una lógica de un nivel elevado.

Para ello, los puntos de características, añaden un componente adicional a los

cinco (EI, EO, EQ, ILF y ELF) ya mencionados de los FP de Albrecht. Este componente denominado algoritmos complejos (CA – del inglés complex algorithms) permite introducir en el conteo a procesos con algoritmos complejos o lógica de nivel elevado. El componente CA por defecto tiene un peso de tres (3). En general el método de puntos de características presenta un peso fijo con respecto a los FP de Albrecht. Para el caso de funciones transaccionales (EI, EO y EQ) se considera una complejidad media; por ende, los valores que toman estos componentes son de cuatro (4), cinco (5) y cuatro (4) respectivamente. Con los grupos de datos lógicos (ILF y ELF) sucede algo similar, pero en vez de considerar los valores de la complejidad media de los FP, considera los de complejidad baja, adjudicándose siete (7) y cinco (5) respectivamente¹¹¹. En el cuadro 26 se muestra la disposición de los componentes con sus respectivos pesos.

Cuadro 70. Pesos para Puntos de Características

COMPONENTE	PESO	CANTIDAD	SUBTOTAL (PESO x CANTIDAD)
ILF	7	---	---
ELF	7	---	---
CA	3	---	---
EI	4	---	---
EO	5	---	---
EQ	4	---	---

Fuente: OVEJERO, José Daniel. Estimación de Proyectos Para Sistemas Basados en Conocimiento [en línea]. Tesis de magister en ingeniería del software. Buenos Aires, Argentina: Instituto Tecnológico de Buenos Aires, 2006. p. 42. [Consultado el 26 de Febrero de 2012]. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Las diferencias entre los FP de Albrecht y los puntos de características, son básicamente que cuando el número de algoritmos de una aplicación es mucho mayor que el de los archivos, los puntos de características generan estimaciones más precisas que los FP. A la inversa si existe un número de archivos considerablemente mayor al de algoritmos (situación muy común en los sistemas de gestión), la cuenta de puntos de características será mucho menos confiable que la de FP.

¹¹¹ Ibíd., p. 40-41. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

En proyectos pequeños, tanto FP de Albrecht como los puntos de características generan los mismos resultados; pero si se realiza el conteo en el marco de proyectos de mediana o gran envergadura, existe una diferencia considerable entre los resultados arrojados por una y otra técnica.

El ajuste de los puntos de características se realiza de igual forma que los FP de Albrecht; es decir se identifican y evalúan los 14 parámetros o características para representar el entorno de desarrollo¹¹².

¹¹² Ibid., p. 42. Disponible en internet: <<http://www.itba.edu.ar/archivos/secciones/ovejero-tesisdemagister.pdf>>

Anexo H. Valor del voltaje y frecuencia dependiendo del país.

#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]	#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]
EUROPA				ÁFRICA			
1	Albania	220	50	79	Algeria	125 / 220	50
2	Alemania	220	50	80	Alto Volta	220	50
3	Austria	220	50	81	Angola	220	50
4	Azores	220	50	82	Botswana	220	50
5	Bélgica	220	50	83	Burkina Faso	220	50
6	Bulgaria	220	50	84	Burma	230	50
7	Checoslovaquia	220	50	85	Burundi	220	50
8	Dinamarca	220	50	86	Camerún	220	50
9	España*	220	50	87	Chad	220	50
10	Estonia	220	50	88	Costa de Marfil	220	50
11	Finlandia	220	50	89	Dahomey	220	50
12	Francia	220	50	90	Djibouti	220	50
13	Gibraltar	240	50	91	Egipto	220	50
14	Grecia	220	50	92	Etiopia	220	50
15	Holanda	220	50	93	Gabón	220	50

* Comprende Islas Baleares, en el mar Mediterráneo; las ciudades de Ceuta y Melilla, en el norte de África y las islas Canarias, en el océano Atlántico.

#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]	#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]
16	Hungría	220	50	94	Gambia	220	50
17	Irlanda	220	50	95	Ghana	220	50
18	Islandia	220	50	96	Guinea Ecuatorial	220	50
19	Italia	220	50	97	Guinea	220	50
20	Latvia	220	50	98	Kenya	240	50
21	Lituania	220	50	99	Liberia	120	60
22	Luxemburgo	220	50	100	Libia	125 / 230	50
23	Malta	240	50	101	Madagascar	220	50
24	Mónaco	220	50	102	Madeira	220	50
25	Noruega	220	50	103	Malagasy (Madagascar)	220	50
26	Polonia	220	50	104	Malawi	230	50
27	Portugal	220	50	105	Mali	220	50
28	Reino Unido	220	50	106	Marruecos	220	50
29	República Eslovaca	220	50	107	Mauricio, Isla	230	50
30	Rumania	220	50	108	Mauritania	220	50
31	Rusia	220	50	109	Mozambique	220	50
32	Suecia	220	50	110	Namibia	220	50
33	Suiza	220	50	111	Niger	220	50
34	Yugoslavia	220	50	112	Nigeria	230	50
AMÉRICA DEL NORTE				113	Reunión, Isla	220	50

#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]	#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]
35	Bahamas	120	60	114	Rwanda	220	50
36	Bermuda	120	60	115	Seychelles, Islas	240	50
37	Canadá	120	60	116	Senegal	220	50
38	Estados Unidos de América	120	60	117	Sierra Leona	230	50
39	Groenlandia	220	50	118	Somalia	110 / 220	50
40	México	125	60	119	Sudáfrica	220 / 250	50
AMÉRICA CENTRAL Y EL CARIBE				120	Sudán	240	50
41	Antillas Holandesas	220	50	121	Swazilandia	230	50
42	Barbados	115	50	122	Tanzania	230	50
43	Costa Rica	120	60	123	Togo	220	50
44	Cuba	120	60	124	Tunisia	220	50
45	El Salvador	115	60	125	Uganda	240	50
46	Guadalupe, Isla	220	50	126	Zaire	220	50
47	Guatemala	120	60	127	Zambia	220	50
48	Honduras	110	60	128	Zimbabwe	220	50
49	Jamaica	110	50	MEDIO ESTE			
50	Martinica, Isla	220	50	129	Arabia Saudita	125 / 220	50 / 60
51	Nicaragua	120	60	130	Chipre	240	50
52	Panamá	120 / 208	60	131	Irán	220	50
53	Puerto rico	120	60	132	Iraq	220	50

#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]	#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]
54	República Dominicana	110	60	133	Israel	230	50
55	San Kitts, Isla	230	60	134	Kuwait	240	50
56	Trinidad-Tobago, Islas	115 / 230	60	135	Líbano	110 / 220	50
AMÉRICA DEL SUR				136	Omán	240	50
57	Argentina	220	50	137	Pakistán	230	50
58	Bolivia	110 / 220	50	138	Qatar	240	50
59	Brasil	110 / 220	60	139	Siria	220	50
60	Chile	220	50	140	Turquía	220	50
61	Colombia	110 / 220	60	141	Yemen	220	50
62	Ecuador	120	60	142	Jordania	220	50
63	Guyana (francesa)	110	50 / 60	ASIA			
64	Paraguay	220	50	143	Afganistán	220	50
65	Perú	110 / 220	50 / 60	144	Bahrain	220	50
66	Surinam	115	60	145	Bangladesh	220	50
67	Uruguay	220	50	146	Brunei	220	50
68	Venezuela	120	60	147	Cambodia	120 / 220	50
AUSTRALIA / ISLAS DEL PACÍFICO				148	China	220	50
69	Australia	240	50	149	Corea del Sur	220	50 / 60
70	Fiji	240	50	150	Filipinas	120	60
71	UAM, Isla	110	60	151	Hong Kong	200	50

#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]	#	PAÍS	TENSIÓN [V]	FRECUENCIA [Hz]
72	Nueva Caledonia	220	50	152	India	220 / 250	50
73	Nueva Guinea	220	50	153	Indonesia	220	50
74	Nueva Zelandia	230	50	154	Japón	100	50 / 60
75	Polinesia Francesa	220	50	155	Malasia	240	50
76	Sabah/Sarawak	220	50	156	Mongolia	220	50
77	Samoa	120	60	157	Nepal	220	50
78	Tahití	220	50	158	Singapur	230	50
				159	Sri Lanka	230	50
				160	Tailandia	220	50
				161	Taiwán	110	60
				162	Vietnam	110 / 220	50

Fuente: asifunciona.com. Tensiones o voltajes y frecuencias de corriente alterna utilizadas por diferentes países. Actualizado Abril del 2007. [Consultado el 07 de Febrero del 2012]. Disponible en internet: <http://www.asifunciona.com/tablas/voltaje_paises/voltaje_paises_2.htm>

Anexo I. Listados de las métricas compatibilidad, comunicación, programación y normas

Compatibilidad		Comunicación	Programación	Normas
Sistema Operativo	Hardware			
Microsoft Windows XP/Vista/7	Robot Motoman SV3X	TCP/IP	Java, JME (Java Micro Edition), EJS	Protocolos Estandarizados
MacOS	Controlador BRC	RTP (Real-Time Transport Protocol)	C, C#, C++	Software Licencia Libre
Linux	Cable DB9	WAP (Wireless Application Protocol)	Objective-C	Sugeridas en "INSTRUCTION MANUAL MOTOMAN, -SV3X, -SV3XL"
Solaris	110/120/220V	GPRS (General Packet Ratio Service)	PHP	Sugeridas en "CONTROLLER-BRC: User Manual"
BlackBerry OS	Frecuencia: 60Hz y 0Hz	UMTS	Visual Basic	Sugeridas en "TEACH PENDANT for CONTROLLER-BRC: User Manual"
Windows Mobile 2003 SE/5.0/6/6.1	Procesador: 166MHz Pentium, 160MHz PowerPC, 166MHz UltraSparc	i-mode	Python	Sugeridas en "ACL-WIN for CONTROLLER-BRC: User Manual"
Windows Phone 6.5/7	RAM > 32MB	.mobi	Perl	Seguridad Industrial

Compatibilidad		Comunicación	Programación	Normas
Sistema Operativo	Hardware			
Symbian	x	RS232	JavaScript	Código Eléctrico Nacional Colombiano – NTC 2050
Android	x	x	ACL WIN	Reglamento Técnico de Instalaciones Eléctricas (RETIE)
iOS	x	x	x	x

Anexo J. Comandos del robot Motoman SPV3

#	Comandos	Función
1	A progame<Enter>	Detiene un programa de usuario en ejecución.
2	A<Enter>	Detiene todos los programas de usuario en ejecución.
3	APPEND<Enter>	Anexa un archivo de respaldo en un archivo residente.
4	ATTACH ?<Enter>	Muestra la lista de todos los vectores de "conectado".
5	ATTACH pvect<Enter>	Conecta vector de posición de TP.
6	ATTACH OFF{A B C}<Enter>	Desconecta vector de TP (del grupo especificado).
7	AUTO <Enter>	Vuelve a activar el modo automático después de salir del modo de aprendizaje.
8	BRAKE axis <Enter>	Libera el freno para ese eje; Utilice COFF para cerrar el freno de nuevo.
9	BREAK linenum <Enter>	Establecer un punto de interrupción en el número de línea;
10	BREAK <Enter>	Hace una lista de todos los puntos de interrupción.
11	NOBREAK linenum <Enter>	Cancela un punto de interrupción en el número de línea;
12	CLOSE<Enter>	Cierra la pinza hasta el final del movimiento de la pinza.
13	CLOSE var<Enter>	Cierra la pinza al DAC especificado (usar con precaución!).
14	CLR n<Enter>	Borra el encoder n o al que resuelve el turno del contador.
15	CLR *<Enter>	Borra todos los encoders o a los que resuelven el turno de los contadores.
16	CLRBUF<Enter>	Vacía el buffer de movimiento de todos los ejes.
17	CLRBUF axis<Enter>	Vacía el buffer de movimiento de un eje específico.
18	CLRBUF{A B}<Enter>	Vacía el buffer de movimiento del grupo A o B.
19	COFF<Enter>	Se desactiva el servo control para todos los grupos y ejes.
20	COFF{A B}<Enter>	Se desactiva el servo control para el grupo A o B solamente.

#	Comandos	Función
21	COFF axis<Enter>	Desconecta el servo control para el 'eje' específico solamente.
22	CON<Enter>	Activa el servo control para todos los grupos y ejes.
23	CON{A B}<Enter>	Activa el servo control para el grupo A o B solamente.
24	CON axis<Enter>	Activa el servo control para el 'eje' específico solamente.
25	CONFIG<Enter>	Abre el archivo de configuración.
26	CONFIG ?<Enter>	Muestra la configuración actual.
27	CONTINUE prognam<Enter>	Reanuda la ejecución de un programa suspendido.
28	COPY prog1 prog2<Enter>	Copia el programa de usuario 'prog1' al nuevo programa de usuario 'prog2'.
29	DEFP{A B} point<Enter>	Crea una posición en el grupo A o B
30	DEFPC point axis<Enter>	Crea una posición en el grupo C para el 'eje' específico.
31	DELETE pvect[index] <Enter>	Borra una posición del vector 'pvect' ubicada en 'index' y empuja las otras posiciones a un lugar más alto.
32	DELP point <Enter>	Borra la posición "point" de la tabla de posición.
33	DELP pvect<Enter>	Borra el vector posición "pvect" de la tabla de posición.
34	DELVAR var<Enter>	Borra la variable "var" de la tabla de variables.
35	DIMG var[n]<Enter>	Define un vector de variables globales de n elementos.
36	DIMP{A B} pvect[n]<Enter>	Define un vector "pvect" que contiene n posiciones para el grupo A o B.
37	DIMPC pvect[n] axis<Enter>	Define un vector "pvect" que contiene n posiciones para un eje en el grupo C.
38	DIR<Enter>	Muestra una lista de todos los programas de usuario.
39	DISABLE {IN OUT} n<Enter>	Da el control manual de ENTRADAS[n] o SALIDAS[n] a través del comando FORCE.
40	DISABLE ?<Enter>	Muestra una lista de todas las E/S 'deshabilitadas'.
41	DO editcomm<Enter>	Ejecuta los comandos de modo EDIT cuando está en el modo DIRECT.
42	DOWLOAD<Enter>	Descarga los parámetros de control para los controladores del servo.

#	Comandos	Función
43	ECHO<Enter>	Habilita el modo ECHO. Muestra todos los caracteres enviados al controlador.
44	EDIT progname<Enter>	Activa el modo EDIT y llama a un programa de usuario.
45	ENGLISH<Enter>	Activa el lenguaje Inglés.
46	edit_cmnd	Comandos que pueden ser ejecutados en un programa de usuario. Cuando esté en el modo DIRECT escriba "DO HELP<Enter>". Cuando esté en el modo EDIT escriba "HELP<Enter>".
47	ENABLE {IN OUT} n<Enter>	Devuelve el control normal del sistema de la entrada o de la salida especificada.
48	EXACT {A B C}<Enter>	El Controlador verifica la llegada de los ejes a la posición correcta antes de ejecutar el siguiente movimiento.
49	EXACT OFF{A B C}<Enter>	El Controlador asume la llegada de los ejes a la posición correcta de acuerdo al tiempo transcurrido.
50	FORCE {IN OUT} n state<Enter>	Activa o desactiva la entrada 'n' o la salida 'n' deshabilitada.
51	FREE<Enter>	Muestra el tamaño del espacio libre en memoria.
52	GACCEL{A B} var<Enter>	Establece un factor de aceleración global, en porcentaje de la aceleración máxima permitida para los ejes del grupo A o B, o todos los grupos si el grupo no se especifica.
53	GACCELC value axis<Enter>	Establece un factor de aceleración global para el eje especificado en el grupo C.
54	GLOBAL variables<Enter>	Define una variable global.
55	GSPEED{A B} var<Enter>	Establece un factor de velocidad global, en porcentaje de la actual velocidad fijada para los ejes del grupo A o B, o todos los grupos si el grupo no se especifica.
56	GSPEEDC value axis<Enter>	Establece un factor de velocidad global para el eje especificado en el grupo C.
57	HELP<Enter>	Ayuda para los comandos en el modo DIRECT/EDIT
58	HERE point<Enter>	Posición de registro: ubicación actual de los ejes en unidades de encoder.
59	HEREC point<Enter>	Posición de registro: ubicación actual de los ejes en unidades cartesianas.
60	HERER point<Enter>	Posición de registro relativa a la posición actual "point" de los ejes, en unidades de encoder.
61	HERERC point2 point1<Enter>	Segunda posición de registro "point2" relativa a la primera "point1", en unidades de encoder. (Coordenadas Cartesianas)

#	Comandos	Función
62	HERERT point2 point1<Enter>	Segunda posición de registro "point2" relativa a la primera "point1", en unidades de encoder. (Coordenadas de Herramienta)
63	HOME<Enter>	Envía todos los ejes del robot a la posición "home" del microswitch.
64	HOME n<Enter>	Envía el eje n a la posición "home" del microswitch.
65	HHOME n<Enter>	Envía el eje n a la posición "home" de hard-stop.
66	INIT EDITOR<Enter>	Inicializa todos los programas de usuario y las variables.
67	INIT CONTROL<Enter>	Inicializa todos los parámetros del sistema.
68	INSERT <i>pvect</i> [<i>index</i>] <Enter>	Inserta una nueva posición en la matriz 'pvect' en el lugar de índice 'index', empujando las posiciones en el índice 'index' a un lugar más alto.
69	JAPANESE<Enter>	Activa el lenguaje Japonés
70	JAW <i>var</i>	Trae la pinza a un porcentaje 'var' de total abierta a la máxima velocidad.
71	JAW <i>var</i> <i>time</i>	Trae la pinza a un porcentaje 'var' de total abierta en el tiempo 'time' especificado.
72	JOINT<Enter>	Establece el modo de movimiento a la Interpolación de Articulación.
73	LET PAR n= <i>val</i> <Enter>	Establece el valor del parámetro n del sistema a 'val'.
74	LINEAR<Enter>	Establece el modo de movimiento a la Interpolación Cartesiana.
75	LIST <Enter>	Muestra todas las líneas de todos los programas de usuario.
76	LIST progname<Enter>	Muestra todas las líneas del programa de usuario especificado 'progname'.
77	LISTP<Enter>	Muestra una lista de todos los puntos de usuario.
78	LISTPV point<Enter>	Muestra las coordenadas de la posición especificada 'point'.
79	LISTVAR<Enter>	Muestra las variables de usuario y del sistema.
80	MODULO <Enter> or MODULO ROLL <Enter>	Redefine la posición actual de Roll en el rango $\pm 360^\circ$
81	MOVE point<Enter>	Mueve el robot a la posición especificada 'point' a la velocidad actual.
82	MOVE point <i>time</i> <Enter>	Mueve el robot a la posición 'point' en el tiempo 'time'.

#	Comandos	Función
83	MOVEC point1 point2<Enter>	Mueve el robot a lo largo de una trayectoria circular desde la posición actual hasta el punto 'point1' a través del punto 'point2'.
84	MOVEL point<Enter>	Mueve el robot a lo largo de una trayectoria lineal desde la posición actual hasta la posición especificada 'point' a la velocidad actual.
85	MOVEL point time<Enter>	Mueve el robot a lo largo de una trayectoria lineal desde la posición actual hasta la posición especificada 'point' en el tiempo 'time'.
86	MOVES pvect firstpoint lastpoint<Enter>	Mueve los ejes sin pausa, a través de cierto número de posiciones de vector consecutivas desde el punto 'firstpoint' hasta el punto 'lastpoint'.
87	MOVES pvect firstpoint lastpoint time<Enter>	Mueve los ejes sin pausa, a través de cierto número de posiciones de vector consecutivas desde el punto 'firstpoint' hasta el punto 'lastpoint' en el tiempo 'time'.
88	NOECHO<Enter>	Desactiva el modo ECHO. Los caracteres enviados al controlador NO son mostrados.
89	NOQUIET<Enter>	Muestra los comandos @DIRECT durante el programa en ejecución.
90	NOSTEP progrname<Enter>	Cancela el modo Paso a Paso del programa.
91	OPEN<Enter>	Abre la pinza hasta el final del movimiento de esta.
92	OPEN var<Enter>	Abre la pinza hasta una DAC especificada 'var'.
93	PANEL <Enter>	Retorna el control al panel físico de switches.
94	PRINT arg1 arg2 ... arg4<Enter>	Muestra las cadenas y variables.
95	PRIORITY progrname value	Establece la prioridad del programa 'progrname' desde 1 hasta 10.
96	QUIET<Enter>	Desactiva ECHO de los comandos @DIRECT
97	RECEIVE progrname<Enter>	Carga el programa de usuario desde un archivo de copia de seguridad.
98	REMOTE <Enter>	Transfiere el control a entradas físicas externas (ver PANEL).
99	REMOVE progrname<Enter>	Borra un programa de usuario.
100	RENAME progrname1 progrname2<Enter>	Cambia el nombre de un programa de usuario de 'progrname1' a 'progrname2'
101	RESETALL <Enter>	Hace un reseteo de hardware del controlador.

#	Comandos	Función
102	RUN progame<Enter>	Ejecuta un programa de usuario 'progame' con la prioridad por defecto.
103	RUN progame priority<Enter>	Ejecuta un programa de usuario 'progame' con la prioridad 'priority'.
104	SEND<Enter>	Envía todos los programas de usuario al host.
105	SEND progame<Enter>	Envía el programa de usuario 'progame' al host.
106	SENDPAR<Enter>	Envía todos los parámetros del sistema al host.
107	SENDPOINT<Enter>	Envía todas las posiciones de usuario al host.
108	SENDPROG<Enter>	Envía todos los programas de usuario al host.
109	SENDVAR<Enter>	Envía todas las variables de usuario al host.
110	SET var1=var2<Enter>	Asigna el valor de 'var2' a 'var1'.
111	SET var1=NOT var2<Enter>	Asigna el valor lógico negativo de 'var2' a 'var1'.
112	SET var1=COMPLEMENT var2<Enter>	Asigna el valor complemento del eje de 'var2' a 'var1'.
113	SET var1=ABS var2<Enter>	Asigna el valor absoluto de 'var2' a 'var1'.
114	SET var1=var2 operator var3<Enter>	Asigna el resultado de una operación aritmética a 'var1'.
115	SET var1=PVAL point axis<Enter>	Asigna el valor de encoder del eje 'axis' en el punto 'point' a 'var1'.
116	SET var1=PVALC point coord<Enter>	Asigna el valor cartesiano de 'coord' en el punto 'point' a 'var1'.
117	SET var1=PAR var2<Enter>	Asigna el valor del parámetro 'var2' a 'var1'.
118	SET var1=PSTATUS point<Enter>	Asigna un valor de acuerdo al estado de la posición especificada 'point'.
119	SETP point1=point2<Enter>	Copia el valor del punto 'point2' a 'point1'.
120	SETPV point<Enter>	Establece un nuevo valor para una posición 'point', en unidades de encoder.
121	SETPV point axis value<Enter>	Establece un nuevo valor para un eje 'axis', en unidades de encoder.
122	SETPVC point coord value<Enter>	Establece un nuevo valor para una coordenada cartesiana 'coord'.
123	SHIFT point BY axis value<Enter>	Desplaza la ubicación del eje por valores de encoder.

#	Comandos	Función
124	SHIFTC point BY coord value<Enter>	Desplaza la posición de la coordenada cartesiana por valores métricos.
125	SHOW DIN<Enter>	Muestra el estado de todas las entradas.
126	SHOW DOUT<Enter>	Muestra el estado de todas las salidas.
127	SHOW ENCO<Enter>	Muestra los valores de todos los encoders cada 0.5 segundos. ^C lo detiene.
128	SHOW DAC n<Enter>	Muestra el valor del DAC para el eje n en milivoltios.
129	SHOW PAR n<Enter>	Muestra el valor del parámetro del sistema.
130	SHOW SPEED<Enter>	Muestra la velocidad actual.
131	SPEED{A B} value<Enter>	Establece la velocidad actual al valor 'value' para el grupo A o B o todos los grupos si el grupo NO es especificado.
132	SPEEDC value axis<Enter>	Establece la velocidad actual al valor 'value' para el eje especificado 'axis' en el grupo C.
133	SPEEDL value<Enter>	Establece la velocidad para movimientos lineales y circulares en mm/s.
134	GSPEED{A B} var<Enter>	Establece un factor de velocidad global, en porcentaje 'var' de la velocidad actualmente fijada para los ejes del grupo A o B, o todos los grupos si el grupo no se especifica.
135	GSPEEDC value axis<Enter>	Establece un factor de velocidad global para
136	SPLINE(L)(D) pvect firstpoint lastpoint<Enter>	Mueve los ejes, sin pausa, cerca de cualquier número de posiciones consecutivas del vector 'pvect', a partir de 'firstpoint' a 'lastpoint'.
137	SPLINE(L)(D) pvect firstpoint lastpoint time<Enter>	Mueve los ejes, sin pausa, cerca de cualquier número de posiciones consecutivas del vector 'pvect', a partir de 'firstpoint' a 'lastpoint', en el tiempo 'time'.
138	STAT[US]<Enter>	Muestra el nombre, prioridad, estado y la línea de comando actual del programa de usuario activo.
139	STEP progname<Enter>	Ejecuta un programa en el modo paso a paso.
140	SUSPEND progname<Enter>	Suspende la ejecución de un programa.
141	TEACH point<Enter>	Establece las coordenadas del punto 'point' desde el teclado, en coordenadas cartesianas.
142	TEACHR point1 point2<Enter>	Establece las coordenadas del punto 'point1' relativo al punto 'point2' desde el teclado, en coordenadas cartesianas.
143	TEACHR point <Enter>	Establece las coordenadas del punto 'point' relativo a la posición actual desde el teclado, en coordenadas cartesianas.

#	Comandos	Función
144	TEACHRT point1 point2<Enter>	Establece las coordenadas del punto 'point1' relativo al punto 'point2' desde el teclado, en coordenadas de herramienta.
145	TEACHRT point <Enter>	Establece las coordenadas del punto 'point' relativo a la posición actual desde el teclado, en coordenadas de herramienta.
146	TOOL length offset angle<Enter>	Define la posición del efector final con respecto a la brida.
147	TON<Enter>	Activa la protección térmica. (¡¡Solamente MK2!!)
148	TON axis<Enter>	Activa la protección térmica para el eje 'axis'. (¡¡Solamente MK2!!)
149	TOFF<Enter>	Desactiva la protección térmica. (¡¡Solamente MK2!!)
150	TOFF axis<Enter>	Desactiva la protección térmica para el eje 'axis'. (¡¡Solamente MK2!!)
151	TRIGGER prog by IN/OUT n state<Enter>	El estado es de 1,0 o se omite. Inicia la ejecución del programa 'prog' cuando la entrada n (o salida) se encuentra en un estado. Si se omite el estado, inicia la ejecución en cada cambio de estado. El programa se ejecuta una vez, y el comando de activación se debe renovado si otra respuesta para la entrada es deseada.
152	UNDEF point<Enter>	Inicializa los valores de posición.
153	UPLOAD<Enter>	Lee los parámetros de control desde los drivers.
154	VER<Enter>	Muestra la información de la versión ACL.
155	ZSET<Enter>	Restablece todos los pulsos de índice del encoder (motores DC).

Anexo K. Java Media Framework (JMF) 2.1.1 – Formatos Media soportados

JMF soporta una tasa de muestreo de audio desde 8KHz hasta 48KHz. Note que la versión multi-plataforma de JMF solamente soporta las siguientes tasas: 8, 11.025, 11.127, 16, 22.05, 22.254, 32, 44.1, y 48 KHz.

La JMF 2.1.1 soporta los tipos de media y formatos listados a bajo en el cuadro 27, en donde, **D** indica que el formato puedes ser decodificado y presentado; **E** indica que el media stream puede ser codificado en el formato; **read** indica que el tipo de media puede ser usado como entrada (leer desde un archivo); **write** indica que el tipo de media puede ser generado como salida (escrito en un archivo):

Cuadro 71. Tipos de Media y Formatos soportados por JMF 2.1.1

Tipo de Media	Versión Multi-plataforma	Solaris/Linux Performance Pack	Windows Performance Pack
AIFF (.aiff)	read/write	read/write	read/write
8-bit mono/stereo linear	D,E	D,E	D,E
16-bit mono/stereo linear	D,E	D,E	D,E
G.711 (U-law)	D,E	D,E	D,E
A-law	D	D	D
IMA4 ADPCM	D,E	D,E	D,E
AVI (.avi)	read/write	read/write	read/write
Audio: 8-bit mono/stereo linear	D,E	D,E	D,E
Audio: 16-bit mono/stereo linear	D,E	D,E	D,E
Audio: DVI ADPCM compressed	D,E	D,E	D,E
Audio: G.711 (U-law)	D,E	D,E	D,E
Audio: A-law	D	D	D
Audio: GSM mono	D,E	D,E	D,E
Audio: ACM [*]	-	-	D,E
Video: Cinepak	D	D,E	D

^{*} Soporte de Administrador de Compresión de Audio de Windows (del inglés Window's Audio Compression Manager support – ACM). Probado para estos formatos: A-law, GSM610, MSNAudio, MSADPCM, Truespeech, mp3, PCM, Voxware AC8, Voxware AC10.

Tipo de Media	Versión Multi-plataforma	Solaris/Linux Performance Pack	Windows Performance Pack
Video: MJPEG (422)	D	D,E	D,E
Video: RGB	D,E	D,E	D,E
Video: YUV	D,E	D,E	D,E
Video: VCM*	-	-	D,E
GSM (.gsm)	read/write	read/write	read/write
GSM mono audio	D,E	D,E	D,E
HotMedia (.mvr)	read	read	Read
IBM HotMedia	D	D	D
MIDI (.mid)	read	read	read
Type 1 & 2 MIDI	-	D	D
MPEG-1 Video (.mpg)	-	read	read
Multiplexed System stream	-	D	D
Video stream	-	D	D
MPEG Layer II Audio (.mp2)	read	read/write	read/write
MPEG layer 1, 2 audio	D	D,E	D,E
QuickTime (.mov)	read/write	read/write	read/write
Audio: 8 bits mono/stereo linear	D,E	D,E	D,E
Audio: 16 bits mono/stereo linear	D,E	D,E	D,E
Audio: G.711 (U-law)	D,E	D,E	D,E
Audio: A-law	D	D	D
Audio: GSM mono	D,E	D,E	D,E
Audio: IMA4 ADPCM	D,E	D,E	D,E
Video: Cinepak	D	D,E	D
Video: H.261	-	D	D
Video: H.263	D	D,E	D,E
Video: JPEG (420, 422, 444)	D	D,E	D,E
Video: RGB	D,E	D,E	D,E

* Soporte de Administrador de Compresión de Vídeo de Windows (del inglés Window's Video Compression Manager support – VCM). Probado para estos formatos: IV41, IV51, VGPX, WINX, YV12, I263, CRAM, MPG4.

Tipo de Media	Versión Multi-plataforma	Solaris/Linux Performance Pack	Windows Performance Pack
Sun Audio (.au)	read/write	read/write	read/write
8 bits mono/stereo linear	D,E	D,E	D,E
16 bits mono/stereo linear	D,E	D,E	D,E
G.711 (U-law)	D,E	D,E	D,E
A-law	D	D	D
Wave (.wav)	read/write	read/write	read/write
8-bit mono/stereo linear	D,E	D,E	D,E
16-bit mono/stereo linear	D,E	D,E	D,E
G.711 (U-law)	D,E	D,E	D,E
A-law	D	D	D
GSM mono	D,E	D,E	D,E
DVI ADPCM	D,E	D,E	D,E
MS ADPCM	D	D	D
ACM	-	-	D,E

Fuente: Oracle. JMF 2.1.1 - Supported Formats. [Consultado el 22 de Marzo del 2012]. Disponible en internet: <http://www.oracle.com/technetwork/java/javase/formats-138492.html>

Anexo L. Formatos RTP soportados

JMF 2.1.1 puede recibir y transmitir los formatos RTP del cuadro 28, en donde, **R** indica que el formato puede ser decodificado y presentado; **T** indica que los media streams pueden ser codificados y transmitidos en el formato:

Cuadro 72. Formatos RTP que puede recibir y transmitir JMF 2.1.1

Tipo de Media	Carga útil RTP	Versión Multi-plataforma	Solaris/Linux Performance Pack	Windows Performance Pack
Audio: G.711 (U-law) 8 kHz	0	R,T	R,T	R,T
Audio: GSM mono	3	R,T	R,T	R,T
Audio: G.723 mono	4	R	R,T	R,T
Audio: 4-bit mono DVI 8 kHz	5	R,T	R,T	R,T
Audio: 4-bit mono DVI 11.025 kHz	16	R,T	R,T	R,T
Audio: 4-bit mono DVI 22.05 kHz	17	R,T	R,T	R,T
Audio: MPEG Layer I, II	14	R,T	R,T	R,T
Video: JPEG (420, 422, 444)*	26	R	R,T	R,T
Video: H.261	31	-	R	R
Video: H.263**	34	Modo A solamente	R,T	R,T
Video: MPEG-I***	32	T	R,T	R,T

Fuente: Oracle. JMF 2.1.1 - Supported Formats. [Consultado el 22 de Marzo del 2012]. Disponible en internet: <http://www.oracle.com/technetwork/java/javase/formats-138492.html>

* JPEG/RTP puede ser transmitido en dimensiones de video que son múltiplos de ocho pixeles, solamente.

** H.263/RTP puede ser transmitido en tres (3) diferentes dimensiones de video: SQCIF (128x96), QCIF (176x144) and CIF (352x288).

*** MPEG/RTP puede ser transmitido solamente desde un contenido MPEG pre-codificado, por ejemplo, desde un archivo MPEG-codificado o desde una fuente de captura habilitada MPEG. La codificación MPEG en tiempo real no es fiable para la transmisión RTP.

Anexo M. Dispositivos de Captura soportados

JMF 2.1.1 soporta los dispositivos de captura SunVideo / SunVideoPlus en Solaris. En Windows, los dispositivos de captura que son soportados tienen controladores VFW (Video For Windows). En Linux, se espera que trabajen los dispositivos que tienen un controlador Video4Linux, pero no han sido ampliamente probados. El cuadro 29 de abajo lista los dispositivos de captura conocidos que trabajan con esta versión.

Cuadro 73. Dispositivos de captura soportados

Dispositivo de Captura	Versión Multi-plataforma	Solaris Performance Pack	Windows Performance Pack
JavaSound (16-bit, 44100, 22050, 11025Hz, 8000Hz linear)	X (J2SE 1.3+)	X	X
SunVideo	-	X	-
SunVideoPlus	-	X	-
VFW	-	-	X
Intel Create & Share	-	-	Win9x
Diamond Supra Video Kit; Share	-	-	Win98
QuickCam VC (camera)	-	-	WinNT
e-cam (camera)	-	-	WinNT, 9X
Winnow Videum	-	-	WinNT, 9X
Creative Web Cam II	-	-	Win9X
Miro Video DC30	-	-	Win9X
Iomega Buz	-	-	Win9X
QuickCam Home USB (Camera)	-	-	Win98
Smart Video Recorder III	-	-	Win9X

Fuente: Oracle. JMF 2.1.1 - Supported Formats. [Consultado el 22 de Marzo del 2012]. Disponible en internet: <http://www.oracle.com/technetwork/java/javase/formats-138492.html>

Anexo N. RXTX License y GNU Lesser General Public License

RXTX License v 2.1 - LGPL v 2.1 + Linking Over Controlled Interface. RXTX is a native interface to serial ports in java. Copyright 1997-2007 by Trent Jarvi tjarvi@qbang.org and others who actually wrote it. See individual source files for more information.

A copy of the LGPL v 2.1 may be found at <http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html> on November 21st 2007 A copy is here for your convenience.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

An executable that contains no derivative of any portion of RXTX, but is designed to work with RXTX by being dynamically linked with it, is considered a "work that uses the Library" subject to the terms and conditions of the GNU Lesser General Public License.

The following has been added to the RXTX License to remove any confusion about linking to RXTX. We want to allow in part what section 5, paragraph 2 of the LGPL does not permit in the special case of linking over a controlled interface. The intent is to add a Java Specification Request or standards body defined interface in the future as another exception but one is not currently available.

<http://www.fsf.org/licenses/gpl-faq.html#LinkingOverControlledInterface>

As a special exception, the copyright holders of RXTX give you permission to link RXTX with independent modules that communicate with RXTX solely through the Sun Microsystems CommAPI interface version 2, regardless of the license terms of these independent modules, and to copy and distribute the resulting combined

work under terms of your choice, provided that every copy of the combined work is accompanied by a complete copy of the source code of RXTX (the version of RXTX used to produce the combined work), being distributed under the terms of the GNU Lesser General Public License plus this exception. An independent module is a module which is not derived from or based on RXTX.

Note that people who make modified versions of RXTX are not obligated to grant this special exception for their modified versions; it is their choice whether to do so. The GNU Lesser General Public License gives permission to release a modified version without this exception; this exception also makes it possible to release a modified version which carries forward this exception.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA All trademarks belong to their respective owners.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to

guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library

is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d

requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under

Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Copyright 1998-2007 Keane Jarvi. For problems or questions regarding this web page contact [Trent Jarvi](#). Last updated: Wed Nov 21 17:34:04 MST 2007

Fuente: JARVI, Trent. RXTX [en línea]. Actualizado 21 de Noviembre del 2007 [Consultado el 10 de Agosto del 2011]. Disponible en internet: <<http://users.frii.com/jarvi/rxtx/license.html>>

Anexo O. Código de Giobot Cliente

```
package giobot;

/*
 * Giovanni Zorrilla Prieto, de ahora en adelante "el autor", NO se hace
 responsable
 * por cualquier daño o perjuicio como resultado del uso, modificación o
 distribución
 * de este Software o sus derivados. En ningún caso el autor será responsable por
 * cualquier pérdida de ingresos, beneficios o datos, o por daños directos,
 * indirectos, especiales, accidentales o punitivos, independientemente de la causa
 * y sin importar la teoría de la responsabilidad, que surja del uso de o
 imposibilidad
 * de uso de este software, incluso si el autor ha sido advertido de la posibilidad
 * de tales daños.
 *
 * Este software no está diseñado ni está destinado para su uso en el control en
 * línea de aeronaves, tráfico aéreo, comunicaciones de navegación o de
 aeronaves,
 * o en el diseño, construcción, operación o mantenimiento de cualquier instalación
 * nuclear. El usuario de este software representa y garantiza que no va a usar
 * o redistribuir este software para esos fines.
 */
public class Giobot extends javax.swing.JFrame {

    public Giobot() {
        super("Giobot");
        initComponents();
        setLocationRelativeTo(null);
        setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                cerrarConexion();
                System.exit(0);
            }
        });
    }

    private void cerrarConexion(){
        establecerCampoTextoEditable(false);
        desconectar.setEnabled(false);
        if (conexionCliente != null){
            if (conexionCliente.isClosed() == false){
```



```

try {
    if (mediaProcesador != null) {
        if (mediaProcesador.getState() != mediaProcesador.Unrealized){
            mostrarMensaje("\nCerrando captura de Video...");
            mediaProcesador.stop();
            mediaProcesador.close();
            mediaProcesador.removeControllerListener(captura);
            captura = null;
            mediaProcesador = null;
            panelVideo.setViewportView(null);
            ALED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/ALEDoff.png")))
;
        }
    }
    mostrarMensaje("\nCerrando conexión Cliente-Servidor...");
    salida.close();
    entrada.close();
    conexionCliente.close();
} catch (java.io.IOException e) {
    mostrarMensaje("\nError - " + e.getMessage());
} catch (NullPointerException e){
    mostrarMensaje("\nError - " + e.getMessage());
}
}
}
VLED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/VLEDoff.png")))
;
dConectar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Conectar.png"))
);
dConectar.setEnabled(true);
dConectar.setToolTipText("Conectar");
conexionCliente = null;
}
private void conectarAServidor() throws java.io.IOException{
    mostrarMensaje("\nConectando...");
    conexionCliente = new
java.net.Socket(java.net.InetAddress.getByName(conexionServidor),
conexionPuerto);
    mostrarMensaje("\nConectado a " +
conexionCliente.getInetAddress().getHostAddress()+"(" +
conexionCliente.getInetAddress().getHostName()+"):"+conexionPuerto);

```

```

    }
    private void dialogo(){
        CPropiedades dialogo = new CPropiedades(this);
        dialogo.setLocationRelativeTo(this);
        dialogo.setVisible(true);
        ok = dialogo.leerOk();
        if(ok){
            conexionServidor = dialogo.leerIP();
            conexionPuerto = dialogo.leerPuerto();
            puertoVideo = dialogo.leerPuertoVideo();
            cerrarConexion();
            ejecutarCliente app = new ejecutarCliente();
        }
        System.runFinalization();
    }
    private class ejecutarCapturaVideo implements
Runnable,javax.media.ControllerListener{
        ejecutarCapturaVideo() {
            Thread ejecutarVideo = new Thread(this, "ejecutarVideo");
            ejecutarVideo.start();
        }
        @SuppressWarnings("static-access")
        public void run(){
            try{
                localizadorVideo = new
javax.media.MediaLocator("rtp://" + conexionServidor + ":" + puertoVideo + "/video");
                mostrarMensaje("\nLocalización Vídeo: " + localizadorVideo);
                fuenteVideoEntrada =
javax.media.Manager.createDataSource(localizadorVideo);
                mostrarMensaje("\nFuente de Vídeo: " + fuenteVideoEntrada);
                mediaProcesador =
javax.media.Manager.createProcessor(fuenteVideoEntrada);
                mediaProcesador.addControllerListener(this);
                mostrarMensaje("\nProcesador de Media: " + mediaProcesador);
                mediaProcesador.configure();
                instante = System.currentTimeMillis();
                while (mediaProcesador.getState() < mediaProcesador.Configured &&
(espera = System.currentTimeMillis() - instante) <= 20000){
                    if (espera >= 20000) {
                        mostrarMensaje("\nNo se pudo configurar el Procesador de Media.");
                        mediaProcesador.stop();
                        mediaProcesador.close();
                        mediaProcesador.removeControllerListener(captura);
                        mediaProcesador = null;
                    }
                }
            }
        }
    }

```

```

    }
    mediaProcesador.realize();
    instante = System.currentTimeMillis();
    while (mediaProcesador.getState() < mediaProcesador.Realized    &&
(espera = System.currentTimeMillis() - instante) <= 20000) {}
    if (espera >= 20000) {
        mostrarMensaje("\nNo se pudo realizar el Procesador de Media.");
        mediaProcesador.stop();
        mediaProcesador.close();
        mediaProcesador.removeControllerListener(captura);
        mediaProcesador = null;
    }
} catch (java.io.IOException e) {
    javax.swing.JOptionPane.showMessageDialog(null,
e.getMessage(), "Giobot Cliente 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
} catch (javax.media.NoProcessorException e) {
    mostrarMensaje("\nNo se pudo crear un Procesador de Media.");
} catch (javax.media.NoDataSourceException e) {
    mostrarMensaje("\nNo se pudo crear una Fuente de Datos.");
} catch (NullPointerException e) {
    mostrarMensaje("\nNo se pudo crear el objeto.");
}
}
}

public void controllerUpdate(javax.media.ControllerEvent ce) {
    if (ce instanceof javax.media.ConfigureCompleteEvent) {
        mostrarMensaje("\nConfigurando el Procesador de Media...");
        pistas = mediaProcesador.getTrackControls();
        if (pistas == null || pistas.length < 1) {
            mostrarMensaje("\nNo se encontró una pista de vídeo.");
        } else {
            mediaProcesador.setContentDescriptor(null);
            for (i = 0; i < pistas.length; i++) {
                mostrarMensaje("\nObteniendo pistas: "+pistas[i]);
                if (pistas[i].isEnabled()) {
                    soportado = pistas[i].getSupportedFormats();
                    if (soportado.length > 0) {
                        if (soportado[0] instanceof javax.media.format.VideoFormat) {
                            elegido = soportado[0];
                            pistas[i].setFormat(elegido);
                        }
                    }
                } else {
                    mostrarMensaje("\nLa pista NO soporta ningun formato.");
                }
            }
        }
    }
}

```

```

        }
    }
}
if (ce instanceof javax.media.RealizeCompleteEvent) {
    mediaProcesador.start();
    if ((componente = mediaProcesador.getVisualComponent())!=null){
        panelVideo.setViewportView(componente);
        ALED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/ALEDon.png")))
;
        } else {
            mostrarMensaje("\nNo se pudo capturar las imágenes de vídeo.");
        }
    }
}
private class ejecutarCliente implements Runnable{
    ejecutarCliente() {
        Thread ejecutar = new Thread(this, "ejecutar");
        ejecutar.start();
    }
    public void run(){
        try {
            dConectar.setEnabled(false);
            desconectar.setEnabled(false);
            conectarAServidor();
            VLED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/VLEDon.png")))
;
            dConectar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Desconectar.png")))
;
            dConectar.setToolTipText("Desconectar");
            obtenerFlujos();
            procesarConexion();
        } catch (java.io.EOFException e) {
            mostrarMensaje("\nError - Se perdió la conexión.");
        } catch (java.io.IOException e) {
            mostrarMensaje("\nError - Conexión cerrada. " + e.getMessage());
        }
        finally {
            cerrarConexion();
        }
    }
}

```

```

    }
    }
}
private void enviaOrden(String mensaje){
    try {
        if (chat.isSelected()){
            salida.writeObject("@"+mensaje);
            salida.flush();
        } else {
            salida.writeObject(mensaje);
            salida.flush();
        }
        mostrarMensaje("\nUsuario> " + mensaje+"\n>");
    } catch (java.io.IOException eIO) {
        mostrarMensaje("\nError al escribir el objeto - " + eIO.getMessage());
    } catch (NullPointerException eNP) {
        mostrarMensaje("\nError - Es posible que NO se haya establecido una
conexión. " + eNP.getMessage());
    }
}
private void establecerCampoTextoEditable(final boolean editable){
    javax.swing.SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            campoTexto.setEditable(editable);
        }
    });
}
public void mostrarMensaje(final String mensajeAMostrar){
    javax.swing.SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            areaTexto.append(""+mensajeAMostrar);
            areaTexto.setCaretPosition(areaTexto.getText().length());
        }
    });
}
private void obtenerFlujos() throws java.io.IOException{
    salida
                                =
                                new
java.io.ObjectOutputStream(conexionCliente.getOutputStream());
    salida.flush();
    entrada = new java.io.ObjectInputStream(conexionCliente.getInputStream());
    mostrarMensaje("\nSe recibieron los flujos de Entrada/Salida.");
}
private void procesarConexion() throws java.io.IOException{
    if (pEmergencia.isSelected() == false){

```

```

        establecerCampoTextoEditable(true);
    }
    do {
        try {
            dConectar.setEnabled(true);
            desconectar.setEnabled(true);
            mensaje = (String) entrada.readObject();
            if (!mensaje.equals("")){
                mostrarMensaje(mensaje);
                if (mensaje.contains("(201)")||mensaje.contains("(72)")){
                    servoON.setSelected(false);
                    servoON.setText("ServoON");
                    servoON.setForeground(java.awt.Color.BLUE);
                }
                if (mensaje.contains("(73)")){
                    servoON.setSelected(true);
                    servoON.setText("ServoOFF");
                    servoON.setForeground(java.awt.Color.RED);
                }
            }
            if (captura==null) captura = new ejecutarCapturaVideo();
        } catch (ClassNotFoundException e) {
            mostrarMensaje("\nSe recibió un objeto de tipo desconocido.");
        }
    } while (!mensaje.equals("DESCONECTAR"));
}
@SuppressWarnings("unchecked")
private void initComponents() {

    servoON = new javax.swing.JToggleButton();
    gripper = new javax.swing.JToggleButton();
    pEmergencia = new javax.swing.JToggleButton();
    campoTexto = new javax.swing.JTextField();
    scrollAreaTexto = new javax.swing.JScrollPane();
    areaTexto = new javax.swing.JTextArea();
    VLED = new javax.swing.JLabel();
    ALED = new javax.swing.JLabel();
    RLED = new javax.swing.JLabel();
    dConectar = new javax.swing.JButton();
    botonNoQuiet = new javax.swing.JToggleButton();
    panelVideo = new javax.swing.JScrollPane();
    chat = new javax.swing.JCheckBox();
    botonIdioma = new javax.swing.JToggleButton();
    botonEco = new javax.swing.JToggleButton();

```

```

botonAyuda = new javax.swing.JButton();
botonMostrarVelocidad = new javax.swing.JButton();
botonMostrarConfiguracion = new javax.swing.JButton();
botonListaProgramas = new javax.swing.JButton();
botonListaLineasProgramas = new javax.swing.JButton();
botonListaPuntos = new javax.swing.JButton();
botonVersionACL = new javax.swing.JButton();
botonAuto = new javax.swing.JButton();
barra = new javax.swing.JMenuBar();
archivo = new javax.swing.JMenu();
abrir = new javax.swing.JMenuItem();
guardar = new javax.swing.JMenuItem();
limpiar = new javax.swing.JMenuItem();
jSeparator1 = new javax.swing.JPopupMenu.Separator();
propiedades = new javax.swing.JMenuItem();
separador = new javax.swing.JPopupMenu.Separator();
desconectar = new javax.swing.JMenuItem();
salir = new javax.swing.JMenuItem();
ayuda = new javax.swing.JMenu();
manuales = new javax.swing.JMenu();
ayudaCtrller = new javax.swing.JMenuItem();
ayudaAClw = new javax.swing.JMenuItem();
ayudaRobot = new javax.swing.JMenuItem();
ayudaTP = new javax.swing.JMenuItem();
about = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Giobot Cliente 1.0.0");
setAutoRequestFocus(false);
setBounds(new java.awt.Rectangle(0, 0, 900, 538));
setFont(new java.awt.Font("Arial", 1, 12));

setIconImage(java.awt.Toolkit.getDefaultToolkit().getImage("src/giobot/Imagenes/
Giobot.png"));
setMinimumSize(new java.awt.Dimension(900, 538));
setName("Giobot");
setResizable(false);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

servoON.setFont(new java.awt.Font("Arial", 1, 8));
servoON.setForeground(new java.awt.Color(0, 0, 255));
servoON.setText("ServoON");
servoON.setMargin(new java.awt.Insets(2, 6, 2, 6));
servoON.setMaximumSize(new java.awt.Dimension(23, 23));

```

```

servoON.setSize(new java.awt.Dimension(23, 23));
servoON.setName("");
servoON.setPreferredSize(new java.awt.Dimension(23, 23));
servoON.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        servoONActionPerformed(evt);
    }
});
getContentPane().add(servoON, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 410, 60, 60));

gripper.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/gCerrado.png"))
);
gripper.setToolTipText("");
gripper.setMaximumSize(new java.awt.Dimension(23, 23));
gripper.setMinimumSize(new java.awt.Dimension(23, 23));
gripper.setPreferredSize(new java.awt.Dimension(23, 23));
gripper.setPressedIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/gAbierto.png")))
;
gripper.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        gripperActionPerformed(evt);
    }
});
getContentPane().add(gripper, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 350, 60, 60));

pEmergencia.setFont(new java.awt.Font("Arial", 0, 12));
pEmergencia.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/paroOFF.png"))
);
pEmergencia.setToolTipText("Detiene todos los programas de usuario que
están siendo ejecutados.");
pEmergencia.setMaximumSize(new java.awt.Dimension(23, 23));
pEmergencia.setMinimumSize(new java.awt.Dimension(23, 23));
pEmergencia.setPreferredSize(new java.awt.Dimension(23, 23));
pEmergencia.setPressedIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/paroON.png")))
;
pEmergencia.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        pEmergenciaActionPerformed(evt);
    }
});

```



```

    });
    getContentPane().add(pEmergencia,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(450, 320, 110, 150));

    campoTexto.setBackground(new java.awt.Color(0, 0, 0));
    campoTexto.setFont(new java.awt.Font("Courier New", 1, 12));
    campoTexto.setForeground(new java.awt.Color(0, 255, 0));
    campoTexto.setToolTipText("");

    campoTexto.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.Bo
rderFactory.createLineBorder(new java.awt.Color(153, 204, 255)), "Comandos",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.BOTTOM, new java.awt.Font("Arial", 1, 10), new
java.awt.Color(255, 255, 255)));
    campoTexto.setCursor(new
java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
    campoTexto.setName("campoTexto");
    campoTexto.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            campoTextoActionPerformed(evt);
        }
    });
    getContentPane().add(campoTexto,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 280, 550, -1));

    scrollAreaTexto.setPreferredSize(new java.awt.Dimension(200, 200));

    areaTexto.setBackground(new java.awt.Color(0, 0, 0));
    areaTexto.setColumns(20);
    areaTexto.setEditable(false);
    areaTexto.setFont(new java.awt.Font("Courier New", 1, 12));
    areaTexto.setForeground(new java.awt.Color(255, 255, 0));
    areaTexto.setLineWrap(true);
    areaTexto.setRows(5);
    areaTexto.setToolTipText("Historial");

    areaTexto.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.Bo
rderFactory.createLineBorder(new java.awt.Color(153, 204, 255)), "Mensajes",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.BOTTOM, new java.awt.Font("Arial", 1, 10), new
java.awt.Color(255, 255, 255)));
    scrollAreaTexto.setViewportViewView(areaTexto);

    getContentPane().add(scrollAreaTexto,                               new

```

```

org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, 550, 260));

    VLED.setFont(new java.awt.Font("Arial", 0, 12));
    VLED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/VLEDoff.png")))
;
    VLED.setToolTipText("Conexión");
    getContentPane().add(VLED,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 320, 20, 20));

    ALED.setFont(new java.awt.Font("Arial", 0, 12));
    ALED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/ALEDoff.png")))
;
    ALED.setToolTipText("Video");
    getContentPane().add(ALED,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 320, 20, 20));

    RLED.setFont(new java.awt.Font("Arial", 0, 12));
    RLED.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/RLEDoff.png")))
;
    RLED.setToolTipText("Paro de Emergencia");
    getContentPane().add(RLED,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 320, 20, 20));

    dConectar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Conectar.png")))
);
    dConectar.setToolTipText("Conectar");
    dConectar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            dConectarActionPerformed(evt);
        }
    });
    getContentPane().add(dConectar,                               new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 350, 60, 60));

    botonNoQuiet.setFont(new java.awt.Font("Arial", 1, 8));
    botonNoQuiet.setForeground(new java.awt.Color(0, 0, 255));
    botonNoQuiet.setText("QUIET");
    botonNoQuiet.setToolTipText("Desactiva el eco de los comandos
@DIRECT.");
    botonNoQuiet.setMargin(new java.awt.Insets(2, 8, 2, 8));

```

```

        botonNoQuiet.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonNoQuietActionPerformed(evt);
            }
        });
        getContentPane().add(botonNoQuiet,
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 410, 60, 60));

        panelVideo.setBackground(new java.awt.Color(0, 0, 0));

        panelVideo.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.B
orderFactory.createLineBorder(new java.awt.Color(153, 204, 255)), "V\u00eddeo",
javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.TOP,
new java.awt.Font("Arial", 1, 12), java.awt.Color.white));
        panelVideo.setForeground(new java.awt.Color(204, 255, 255));
        panelVideo.setToolTipText("Pantalla de V\u00eddeo");

        panelVideo.setViewportBorder(javax.swing.BorderFactory.createBevelBorder(java
x.swing.border.BevelBorder.LOWERED));
        panelVideo.setAutoscrolls(true);
        panelVideo.setCursor(new
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        panelVideo.setFont(new java.awt.Font("Arial", 0, 12));
        getContentPane().add(panelVideo,
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 10, 510, 460));

        chat.setFont(new java.awt.Font("Arial", 1, 12));
        chat.setText("Chat");
        getContentPane().add(chat,
org.netbeans.lib.awtextra.AbsoluteConstraints(390, 320, -1, -1));

        botonIdioma.setFont(new java.awt.Font("Arial", 1, 8));
        botonIdioma.setForeground(new java.awt.Color(255, 0, 0));
        botonIdioma.setText("JAPANESE");
        botonIdioma.setToolTipText("Idioma para los mensajes del robot.");
        botonIdioma.setMargin(new java.awt.Insets(1, 4, 1, 4));
        botonIdioma.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonIdiomaActionPerformed(evt);
            }
        });
        getContentPane().add(botonIdioma,
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 410, 60, 60));

```

```

        botonEco.setFont(new java.awt.Font("Arial", 1, 8));
        botonEco.setForeground(new java.awt.Color(255, 0, 0));
        botonEco.setText("NOECHO");
        botonEco.setToolTipText("Deshabilita el modo ECHO. Los caracteres
enviados al robot NO son mostrados.");
        botonEco.setMargin(new java.awt.Insets(1, 4, 1, 4));
        botonEco.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonEcoActionPerformed(evt);
            }
        });
        getContentPane().add(botonEco, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 350, 60, 60));

        botonAyuda.setFont(new java.awt.Font("Arial", 1, 18));
        botonAyuda.setForeground(new java.awt.Color(0, 153, 51));
        botonAyuda.setText("?");
        botonAyuda.setToolTipText("Ayuda");
        botonAyuda.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonAyudaActionPerformed(evt);
            }
        });
        getContentPane().add(botonAyuda, new
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 350, 60, 60));

        botonMostrarVelocidad.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/cronometro.png
")));
        botonMostrarVelocidad.setToolTipText("Muestra la velocidad actual.");
        botonMostrarVelocidad.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonMostrarVelocidadActionPerformed(evt);
            }
        });
        getContentPane().add(botonMostrarVelocidad, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 350, 60, 60));

        botonMostrarConfiguracion.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/articulacion.png
")));
        botonMostrarConfiguracion.setToolTipText("Muestra la configuración
actual.");

```

```

        botonMostrarConfiguracion.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonMostrarConfiguracionActionPerformed(evt);
            }
        });
        getContentPane().add(botonMostrarConfiguracion,
new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 410, 60, 60));

        botonListaProgramas.setFont(new java.awt.Font("Arial", 1, 12));
        botonListaProgramas.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/programacion.p
ng"))));
        botonListaProgramas.setToolTipText("Muestra una lista de todos los
programas de usuario.");
        botonListaProgramas.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonListaProgramasActionPerformed(evt);
            }
        });
        getContentPane().add(botonListaProgramas,
new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 350, 60, 60));

        botonListaLineasProgramas.setFont(new java.awt.Font("Arial", 1, 12));
        botonListaLineasProgramas.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Lista.png"))));
        botonListaLineasProgramas.setToolTipText("Muestra una lista de todas las
líneas de todos los programas de usuario.");
        botonListaLineasProgramas.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonListaLineasProgramasActionPerformed(evt);
            }
        });
        getContentPane().add(botonListaLineasProgramas,
new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 410, 60, 60));

        botonListaPuntos.setFont(new java.awt.Font("Arial", 1, 12));
        botonListaPuntos.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/ListaPuntos.png
"))));
        botonListaPuntos.setToolTipText("Muestra una lista de todos los puntos de
usuario.");
        botonListaPuntos.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botonListaPuntosActionPerformed(evt);
        }
    });
    getContentPane().add(botonListaPuntos,
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 410, 60, 60));

    botonVersionACL.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/robot.png")));
    botonVersionACL.setToolTipText("Muestra información de la versión de
ACL.");
    botonVersionACL.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botonVersionACLActionPerformed(evt);
        }
    });
    getContentPane().add(botonVersionACL,
org.netbeans.lib.awtextra.AbsoluteConstraints(380, 410, 60, 60));

    botonAuto.setFont(new java.awt.Font("Arial", 1, 10));
    botonAuto.setForeground(new java.awt.Color(0, 0, 255));
    botonAuto.setText("AUTO");
    botonAuto.setToolTipText("Vuelve a activar el modo automático después de
salir del modo de aprendizaje.");
    botonAuto.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botonAutoActionPerformed(evt);
        }
    });
    getContentPane().add(botonAuto,
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 350, 60, 60));

    archivo.setText("Archivo");
    archivo.setToolTipText("");
    archivo.setFont(new java.awt.Font("Arial", 1, 12));

    abrir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent
.VK_A, java.awt.event.InputEvent.CTRL_MASK));
    abrir.setFont(new java.awt.Font("Arial", 0, 12));
    abrir.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/abrir.png")));
    abrir.setText("Abrir");
    abrir.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            abrirActionPerformed(evt);
        }
    });
    archivo.add(abrir);

```

```

    guardar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_G, java.awt.event.InputEvent.CTRL_MASK));
    guardar.setFont(new java.awt.Font("Arial", 0, 12));
    guardar.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/guardar.png")));
    guardar.setText("Guardar");
    guardar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            guardarActionPerformed(evt);
        }
    });
    archivo.add(guardar);

```

```

    limpiar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_L, java.awt.event.InputEvent.CTRL_MASK));
    limpiar.setFont(new java.awt.Font("Arial", 0, 12));
    limpiar.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/historial.png")));
    limpiar.setText("Limpiar Historial");
    limpiar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            limpiarActionPerformed(evt);
        }
    });
    archivo.add(limpiar);
    archivo.add(jSeparator1);

```

```

    propiedades.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_P, java.awt.event.InputEvent.CTRL_MASK));
    propiedades.setFont(new java.awt.Font("Arial", 0, 12));
    propiedades.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/propiedades.png")));
    propiedades.setText("Propiedades...");
    propiedades.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            propiedadesActionPerformed(evt);
        }
    });
    archivo.add(propiedades);
    archivo.add(separador);

    desconectar.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D, java.awt.event.InputEvent.CTRL_MASK));
    desconectar.setFont(new java.awt.Font("Arial", 0, 12));
    desconectar.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Desconectar.png")));
    desconectar.setText("Desconectar");
    desconectar.setDisabledIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Desconectar.png")));
    desconectar.setEnabled(false);
    desconectar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            desconectarActionPerformed(evt);
        }
    });
    archivo.add(desconectar);

    salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4, java.awt.event.InputEvent.ALT_MASK));
    salir.setFont(new java.awt.Font("Arial", 0, 12));
    salir.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/salida.png")));
    salir.setText("Salir");
    salir.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            salirActionPerformed(evt);
        }
    });
    archivo.add(salir);

    barra.add(archivo);

    ayuda.setText("Ayuda");
    ayuda.setFont(new java.awt.Font("Arial", 1, 12));

```



```

        manuales.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/manuales.png")
));
        manuales.setText("Manuales");
        manuales.setToolTipText("");
        manuales.setFont(new java.awt.Font("Arial", 0, 12));

ayudaCtrlr.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.K
eyEvent.VK_N, java.awt.event.InputEvent.CTRL_MASK));
        ayudaCtrlr.setFont(new java.awt.Font("Arial", 0, 12));
        ayudaCtrlr.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/manualControla
dor.png"))));
        ayudaCtrlr.setText("Controlador");
        ayudaCtrlr.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ayudaCtrlrActionPerformed(evt);
            }
        });
        manuales.add(ayudaCtrlr);

ayudaACLw.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.K
eyEvent.VK_M, java.awt.event.InputEvent.CTRL_MASK));
        ayudaACLw.setFont(new java.awt.Font("Arial", 0, 12));
        ayudaACLw.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/manualProgram
acion.png"))));
        ayudaACLw.setText("Programación");
        ayudaACLw.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ayudaACLwActionPerformed(evt);
            }
        });
        manuales.add(ayudaACLw);

ayudaRobot.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.K
eyEvent.VK_O, java.awt.event.InputEvent.CTRL_MASK));
        ayudaRobot.setFont(new java.awt.Font("Arial", 0, 12));
        ayudaRobot.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/manualRobot.p

```

```

ng"))));
    ayudaRobot.setText("Robot");
    ayudaRobot.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ayudaRobotActionPerformed(evt);
        }
    });
    manuales.add(ayudaRobot);

    ayudaTP.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_T, java.awt.event.InputEvent.CTRL_MASK));
    ayudaTP.setFont(new java.awt.Font("Arial", 0, 12));
    ayudaTP.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/TP.png")));
    ayudaTP.setText("Teach Pendant");
    ayudaTP.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ayudaTPActionPerformed(evt);
        }
    });
    manuales.add(ayudaTP);
    ayuda.add(manuales);

    about.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_Y, java.awt.event.InputEvent.CTRL_MASK));
    about.setFont(new java.awt.Font("Arial", 0, 12));
    about.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Giobot.png")));
    about.setText("Acerca de...");
    about.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            aboutActionPerformed(evt);
        }
    });
    ayuda.add(about);
    barra.add(ayuda);
    setJMenuBar(barra);
    pack();
}

private void campoTextoActionPerformed(java.awt.event.ActionEvent evt) {
    enviaOrden(evt.getActionCommand());
}

```

```

        campoTexto.setText("");
    }

    private void pEmergenciaActionPerformed(java.awt.event.ActionEvent evt) {
        if (pEmergencia.isSelected()){
            pEmergencia.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/paroON.png")));
            pEmergencia.setPressedIcon(new
            javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/paroOFF.png"))
            );
            RLED.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/RLEDOn.png")))
            ;
            enviaOrden("a");
        }else{
            pEmergencia.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/paroOFF.png"))
            );
            pEmergencia.setPressedIcon(new
            javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/paroON.png")));
            RLED.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/RLEDOff.png")))
            ;
            campoTexto.setText("continue ");
        }
    }

    private void aboutActionPerformed(java.awt.event.ActionEvent evt) {
        CAcercaDe acerca = new CAcercaDe(this);
        acerca.setLocationRelativeTo(this);
        acerca.setVisible(true);
    }

    private void servoONActionPerformed(java.awt.event.ActionEvent evt) {
        if (servoON.isSelected()){
            enviaOrden("con");
            servoON.setText("ServoOFF");
            servoON.setForeground(java.awt.Color.RED);
        }else{
            servoON.setText("ServoON");
            servoON.setForeground(java.awt.Color.BLUE);
            enviaOrden("coff");
        }
    }
}

```

```

private void gripperActionPerformed(java.awt.event.ActionEvent evt) {
    if (gripper.isSelected()){
        gripper.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/gAbierto.png")))
;
        gripper.setPressedIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/gCerrado.png"))
);
        enviaOrden("open");
        gripper.setToolTipText("Cerrar Gripper");
    }else{
        gripper.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/gCerrado.png"))
);
        gripper.setPressedIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/gAbierto.png")))
;
        enviaOrden("close");
        gripper.setToolTipText("Abrir Gripper");
    }
}

private void desconectarActionPerformed(java.awt.event.ActionEvent evt) {
    cerrarConexion();
}

private void propiedadesActionPerformed(java.awt.event.ActionEvent evt) {
    dialogo();
}

private void limpiarActionPerformed(java.awt.event.ActionEvent evt) {
    areaTexto.setText("");
}

private void salirActionPerformed(java.awt.event.ActionEvent evt) {
    cerrarConexion();
    System.exit(0);
}

private void guardarActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        archivosTXT = new archivosTXT();
        archivosTXT.guardarArchivoTXT(this, areaTexto.getText());
    } catch (java.io.FileNotFoundException e){

```

```

        mostrarMensaje("\nNo se pudo encontrar el archivo para guardar.");
    } catch (java.io.IOException e){
        mostrarMensaje("\nNo se pudo leer el archivo para guardar.");
    }
}

private void ayudaCtrllerActionPerformed(java.awt.event.ActionEvent evt) {
    abrirArchivoDeAyuda("src\\giobot\\Manuales\\100352-b.pdf");
}

private void ayudaACLwActionPerformed(java.awt.event.ActionEvent evt) {
    abrirArchivoDeAyuda("src\\giobot\\Manuales\\100354-a.pdf");
}

private void ayudaRobotActionPerformed(java.awt.event.ActionEvent evt) {
    abrirArchivoDeAyuda("src\\giobot\\Manuales\\100355-a.pdf");
}

private void ayudaTPActionPerformed(java.awt.event.ActionEvent evt) {
    abrirArchivoDeAyuda("src\\giobot\\Manuales\\100353-a.pdf");
}

private void dConectarActionPerformed(java.awt.event.ActionEvent evt) {
    if (conexionCliente == null) {
        dialogo();
    } else {
        cerrarConexion();
    }
}

private void botonNoQuietActionPerformed(java.awt.event.ActionEvent evt) {
    if (botonNoQuiet.isSelected()){
        enviaOrden("quiet");
        botonNoQuiet.setText("NOQUIET");
        botonNoQuiet.setToolTipText("Muestra los comandos @DIRECT durante
la ejecución del programa.");
        botonNoQuiet.setForeground(java.awt.Color.RED);
    }else{
        botonNoQuiet.setText("QUIET");
        botonNoQuiet.setToolTipText("Desactiva el eco de los comandos
@DIRECT.");
        botonNoQuiet.setForeground(java.awt.Color.BLUE);
        enviaOrden("noquiet");
    }
}
}

```

```

private void abrirActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        archivosTXT = new archivosTXT();
        mostrarMensaje(archivosTXT.abrirArchivoTXT(this));
    } catch (java.io.FileNotFoundException e){
        mostrarMensaje("No se pudo encontrar el archivo solicitado.");
    } catch (java.io.IOException e){
        mostrarMensaje("\nError - " + e.getMessage()+". NO se puede abrir el
archivo.");
    }
}

private void botonIdiomaActionPerformed(java.awt.event.ActionEvent evt) {
    if (botonIdioma.isSelected()){
        enviaOrden("japanese");
        botonIdioma.setText("ENGLISH");
        botonIdioma.setForeground(java.awt.Color.BLUE);
    }else{
        botonIdioma.setText("JAPANESE");
        botonIdioma.setForeground(java.awt.Color.RED);
        enviaOrden("english");
    }
}

private void botonEcoActionPerformed(java.awt.event.ActionEvent evt) {
    if (botonEco.isSelected()){
        enviaOrden("noecho");
        botonEco.setText("ECHO");
        botonEco.setForeground(java.awt.Color.BLUE);
        botonEco.setToolTipText("Habilita el modo ECHO. Los caracteres
enviados al robot son mostrados.");
    }else{
        botonEco.setText("NOECHO");
        botonEco.setForeground(java.awt.Color.RED);
        botonEco.setToolTipText("Deshabilita el modo ECHO. Los caracteres
enviados al robot NO son mostrados.");
        enviaOrden("echo");
    }
}

private void botonAyudaActionPerformed(java.awt.event.ActionEvent evt) {
    enviaOrden("help");
}

private void botonMostrarVelocidadActionPerformed(java.awt.event.ActionEvent
evt) {

```

```

        enviaOrden("show speed");
    }

    private void botonMostrarConfiguracionActionPerformed(java.awt.event.ActionEvent evt) {
        enviaOrden("config ?");
    }

    private void botonListaProgramasActionPerformed(java.awt.event.ActionEvent
    evt) {
        enviaOrden("dir");
    }

    private void botonListaLineasProgramasActionPerformed(java.awt.event.ActionEvent evt) {
        enviaOrden("list");
    }

    private void botonListaPuntosActionPerformed(java.awt.event.ActionEvent evt) {
        enviaOrden("listp");
    }

    private void botonVersionACLActionPerformed(java.awt.event.ActionEvent evt) {
        enviaOrden("ver");
    }

    private void botonAutoActionPerformed(java.awt.event.ActionEvent evt) {
        enviaOrden("auto");
    }

    private void abrirArchivoDeAyuda(String rutaArchivo){
        try{
            if (java.awt.Desktop.isDesktopSupported()){
                java.awt.Desktop.getDesktop().open(new java.io.File(rutaArchivo));
            } else {
                mostrarMensaje("\nGiobot NO puede abrir el archivo en este Sistema
Operativo.");
            }
        } catch(java.io.IOException e){
            mostrarMensaje("\nError - " + e.getMessage()+". El archivo especificado
NO tiene una aplicación asociada o la aplicación falló al ejecutarse.");
        }
    }
}

```

```

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Windows".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Giobot.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Giobot.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Giobot.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Giobot.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Giobot().setVisible(true);
            }
        });
    }
    private javax.swing.JLabel ALED;
    private javax.swing.JLabel RLED;
    private javax.swing.JLabel VLED;
    private javax.swing.JMenuItem about;
    private javax.swing.JMenuItem abrir;
    private javax.swing.JMenu archivo;
    private javax.swing.JTextArea areaTexto;
    private javax.swing.JMenu ayuda;
    private javax.swing.JMenuItem ayudaACLw;
    private javax.swing.JMenuItem ayudaCtrller;
    private javax.swing.JMenuItem ayudaRobot;
    private javax.swing.JMenuItem ayudaTP;

```



```

private javax.swing.JMenuBar barra;
private javax.swing.JButton botonAuto;
private javax.swing.JButton botonAyuda;
private javax.swing.JToggleButton botonEco;
private javax.swing.JToggleButton botonIdioma;
private javax.swing.JButton botonListaLineasProgramas;
private javax.swing.JButton botonListaProgramas;
private javax.swing.JButton botonListaPuntos;
private javax.swing.JButton botonMostrarConfiguracion;
private javax.swing.JButton botonMostrarVelocidad;
private javax.swing.JToggleButton botonNoQuiet;
private javax.swing.JButton botonVersionACL;
private javax.swing.JTextField campoTexto;
private javax.swing.JCheckBox chat;
private javax.swing.JButton dConectar;
private javax.swing.JMenuItem desconectar;
private javax.swing.JToggleButton gripper;
private javax.swing.JMenuItem guardar;
private javax.swing.JPopupMenu.Separator jSeparator1;
private javax.swing.JMenuItem limpiar;
private javax.swing.JMenu manuales;
private javax.swing.JToggleButton pEmergencia;
private javax.swing.JScrollPane panelVideo;
private javax.swing.JMenuItem propiedades;
private javax.swing.JMenuItem salir;
private javax.swing.JScrollPane scrollAreaTexto;
private javax.swing.JPopupMenu.Separator separador;
private javax.swing.JToggleButton servoON;
private archivosTXT archivosTXT;
private boolean ok;
private int conexionPuerto=0, i, puertoVideo;
private long espera, instante;
private java.awt.Component componente;
private java.io.ObjectOutputStream salida;
private java.io.ObjectInputStream entrada;
private java.net.Socket conexionCliente;
private String conexionServidor, mensaje = "";
private javax.media.Processor mediaProcesador=null;
private javax.media.MediaLocator localizadorVideo;
private javax.media.protocol.DataSource fuenteVideoEntrada;
private javax.media.control.TrackControl[] pistas;
private javax.media.Format soportado[], elegido;
private ejecutarCapturaVideo captura;
}

```

Anexo P. Código de CPropiedades

```
package giobot;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class CPropiedades extends javax.swing.JDialog {
    public CPropiedades(java.awt.Frame padre) {
        super(padre, true);
        initComponents();
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                dispose();
            }
        });
    }
    @SuppressWarnings("unchecked")

    private void initComponents() {

        aceptar = new javax.swing.JButton();
        cancelar = new javax.swing.JButton();
        jTabbedPane1 = new javax.swing.JTabbedPane();
        clienteServidor = new javax.swing.JPanel();
        etiquetaIP = new javax.swing.JLabel();
        etiquetaPuerto = new javax.swing.JLabel();
        ip = new javax.swing.JTextField();
        puerto = new javax.swing.JTextField();
        predeterminado = new javax.swing.JButton();
        etiquetaPuertoVideo = new javax.swing.JLabel();
        campoPuertoVideo = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Propiedades de Conexión");
        setLocationByPlatform(true);
        setMinimumSize(new java.awt.Dimension(238, 238));
        setModal(true);
        setResizable(false);

        aceptar.setText("Aceptar");
        aceptar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        aceptarActionPerformed(evt);
    }
});

cancelar.setText("Cancelar");
cancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cancelarActionPerformed(evt);
    }
});

jTabbedPane1.setFont(new java.awt.Font("Arial", 1, 12));

clienteServidor.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));

etiquetaIP.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaIP.setText("IP :");

etiquetaPuerto.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaPuerto.setText("Puerto :");

ip.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N
ip.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
ip.setText("127.0.0.1");
ip.setToolTipText("###.###.###.###");
ip.setMinimumSize(new java.awt.Dimension(26, 21));
ip.setPreferredSize(new java.awt.Dimension(100, 21));
ip.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ipActionPerformed(evt);
    }
});
ip.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        ipKeyTyped(evt);
    }
});

puerto.setFont(new java.awt.Font("Arial", 0, 12));
puerto.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
puerto.setText("60002");
puerto.setToolTipText("49152 – 65535");

```

```

puerto.setPreferredSize(new java.awt.Dimension(100, 20));
puerto.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        puertoActionPerformed(evt);
    }
});
puerto.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        puertoKeyTyped(evt);
    }
});

predeterminado.setText("Predeterminado");
predeterminado.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        predeterminadoActionPerformed(evt);
    }
});

etiquetaPuertoVideo.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaPuertoVideo.setText("Puerto Video :");

campoPuertoVideo.setFont(new java.awt.Font("Arial", 0, 12));
campoPuertoVideo.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
campoPuertoVideo.setText("60004");
campoPuertoVideo.setToolTipText("49152 – 65535");
campoPuertoVideo.setMinimumSize(new java.awt.Dimension(26, 21));
campoPuertoVideo.setPreferredSize(new java.awt.Dimension(100, 21));
campoPuertoVideo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        campoPuertoVideoActionPerformed(evt);
    }
});
campoPuertoVideo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        campoPuertoVideoKeyTyped(evt);
    }
});

javax.swing.GroupLayout clienteServidorLayout = new
javax.swing.GroupLayout(clienteServidor);
clienteServidor.setLayout(clienteServidorLayout);
clienteServidorLayout.setHorizontalGroup(

```

```

clienteServidorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(clienteServidorLayout.createSequentialGroup()
        .addContainerGap()

        .addGroup(clienteServidorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(etiquetaPuerto)
            .addComponent(etiquetaIP)
            .addComponent(etiquetaPuertoVideo))
            .addGap(10, 10, 10)

        .addGroup(clienteServidorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(campoPuertoVideo,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(puerto, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(ip, javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                clienteServidorLayout.createSequentialGroup()
                    .addContainerGap(90, Short.MAX_VALUE)
                    .addComponent(predeterminado)
                    .addContainerGap())
        );
clienteServidorLayout.setVerticalGroup(

clienteServidorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(clienteServidorLayout.createSequentialGroup()
        .addContainerGap()

        .addGroup(clienteServidorLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(etiquetaIP)
            .addComponent(ip, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```



```

        .addComponent(aceptar)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cancelar)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jTabbedPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,          167,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cancelar)
            .addComponent(aceptar))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
}

private void aceptarActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        if      (Integer.parseInt(puerto.getText())      >=      49152      &&
Integer.parseInt(puerto.getText()) <= 65535){
            puertoValido = true;
        }else{
            puertoValido = false;
            getToolkit().beep();
            javax.swing.JOptionPane.showMessageDialog(null, "Dígame un valor para
el puerto en el intervalo de 49152 hasta 65535.", "Giobot Cliente 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
        }
        if(Integer.parseInt(campoPuertoVideo.getText()) >= 49152 &&
Integer.parseInt(campoPuertoVideo.getText()) <= 65535 &&
Integer.parseInt(campoPuertoVideo.getText()) % 2 == 0){
            validoV = true;
        }else{
            validoV = false;
        }
    }
}

```

```

        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Digite un valor PAR
para el puerto Vídeo en el intervalo de 49152 hasta 65535.", "Giobot Cliente 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if(Integer.parseInt(campoPuertoVideo.getText()) !=
Integer.parseInt(puerto.getText()) &&
        Integer.parseInt(puerto.getText()) !=
(Integer.parseInt(campoPuertoVideo.getText()+1)){
        validol = true;
    }else{
        validol = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor
diferente para Puerto que NO sea consecutivo del puerto para video.", "Giobot
Cliente 1.0.0", javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (puertoValido && validol && validoV){
        Puerto = Integer.parseInt(puerto.getText());
        dir = ip.getText();
        puertoVideo = Integer.parseInt(campoPuertoVideo.getText());
        ok = true;
        dispose();
    } else{
        ok = false;
    }
} catch(NumberFormatException e){
    getToolkit().beep();
    javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor para
los puertos en el intervalo de 49152 hasta 65535.", "Giobot Cliente 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
}
}

private void cancelarActionPerformed(java.awt.event.ActionEvent evt) {
    ok = false;
    dispose();
}

private void campoPuertoVideoKeyTyped(java.awt.event.KeyEvent evt) {
    if (!(Character.isDigit(evt.getKeyChar()) || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_BACK_SPACE || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_DELETE || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_ENTER) || campoPuertoVideo.getText().length() >=
5) {

```



```

        getToolkit().beep();
        evt.consume();
    }
}
private void campoPuertoVideoActionPerformed(java.awt.event.ActionEvent evt)
{
    campoPuertoVideo.transferFocus();
}

private void puertoKeyTyped(java.awt.event.KeyEvent evt) {
    if (!(Character.isDigit(evt.getKeyChar()) || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_BACK_SPACE || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_DELETE || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_ENTER) || puerto.getText().length() >= 5) {
        getToolkit().beep();
        evt.consume();
    }
}

private void puertoActionPerformed(java.awt.event.ActionEvent evt) {
    puerto.transferFocus();
    predeterminado.transferFocus();
}

private void ipKeyTyped(java.awt.event.KeyEvent evt) {
    if (!(Character.isDigit(evt.getKeyChar()) || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_BACK_SPACE || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_DELETE || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_ENTER || evt.getKeyChar() == '.') ||
ip.getText().length() >= 15) {
        getToolkit().beep();
        evt.consume();
    }
}

private void ipActionPerformed(java.awt.event.ActionEvent evt) {
    ip.transferFocus();
}

private void predeterminadoActionPerformed(java.awt.event.ActionEvent evt) {
    campoPuertoVideo.setText("60004");
    ip.setText("127.0.0.1");
    puerto.setText("60002");
}

public String leerIP(){
    return dir;
}

public int leerPuerto(){

```

```

        return Puerto;
    }
    public int leerPuertoVideo(){
        return puertoVideo;
    }
    public boolean leerOk(){
        return ok;
    }
    private javax.swing.JButton aceptar;
    private javax.swing.JTextField campoPuertoVideo;
    private javax.swing.JButton cancelar;
    private javax.swing.JPanel clienteServidor;
    private javax.swing.JLabel etiquetaIP;
    private javax.swing.JLabel etiquetaPuerto;
    private javax.swing.JLabel etiquetaPuertoVideo;
    private javax.swing.JTextField ip;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JButton predeterminado;
    private javax.swing.JTextField puerto;

    private boolean ok = false, puertoValido=false, validoV=false, validoI=false;
    private String dir = "";
    private int Puerto = 0, puertoVideo=0;
}

```

Anexo Q. Código de CAcercaDe

```
package giobot;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class CAcercaDe extends javax.swing.JDialog {
    public CAcercaDe(java.awt.Frame parent) {
        super(parent, true);
        initComponents();
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                dispose();
            }
        });
    }
    @SuppressWarnings("unchecked")

    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        autor = new javax.swing.JLabel();
        profesion = new javax.swing.JLabel();
        universidad = new javax.swing.JLabel();
        version = new javax.swing.JLabel();
        icono = new javax.swing.JLabel();
        cerrar = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Giobot Cliente 1.0.0");
        setBackground(java.awt.Color.black);
        setForeground(new java.awt.Color(204, 204, 204));

        setIconImage(java.awt.Toolkit.getDefaultToolkit().getImage("src/Giobot/Imagenes/Giobot.png"));
        setModal(true);
        setResizable(false);

        jPanel1.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2,
            new java.awt.Color(255, 204, 0)));

        autor.setFont(new java.awt.Font("Tahoma", 1, 11));
```

```

autor.setText("Autor: Giovanni Zorrilla Prieto");

profesion.setFont(new java.awt.Font("Tahoma", 1, 11));
profesion.setText("Ingeniero Mecatrónico");

universidad.setFont(new java.awt.Font("Tahoma", 1, 11));
universidad.setText("Universidad Autónoma de Occidente");
universidad.setToolTipText("");

version.setFont(new java.awt.Font("Tahoma", 1, 11));
version.setText("Versión del producto: Giobot Cliente 1.0.0");

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(profesion, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
        .addComponent(autor)
        .addComponent(universidad,
javax.swing.GroupLayout.DEFAULT_SIZE, 240, Short.MAX_VALUE)
        .addComponent(version, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
        .addGap())
    );
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(autor)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(profesion)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(universidad)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(version)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    icono.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Giobot/Imagenes/Giobot-
1.png"))));

    cerrar.setText("Cerrar");
    cerrar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            cerrarActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup()
                        .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(cerrar))
                    .addComponentGap())
                .addGroup(layout.createSequentialGroup()
                    .addComponentGap(48, Short.MAX_VALUE)
                    .addComponent(icono)
                    .addComponentGap(46, 46, 46))
            );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addComponent(icono)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(cerrar)
        .addContainerGap()
    );

    icono.getAccessibleContext().setAccessibleName("Icono");

    pack();
}

private void cerrarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private javax.swing.JLabel autor;
private javax.swing.JButton cerrar;
private javax.swing.JLabel icono;
private javax.swing.JPanel jPanel1;
private javax.swing.JLabel profesion;
private javax.swing.JLabel universidad;
private javax.swing.JLabel version;
}

```

Anexo R. Código de archivosTXT

```
package giobot;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class archivosTXT {
    private javax.swing.JFileChooser selectorDeArchivo;
    private int seleccion;
    private java.io.File archivo;
    private java.io.PrintWriter guardar;
    private java.io.BufferedReader Abrir;

    public void abrirCerrarArchivosTXT(){}
    public String abrirArchivoTXT(javax.swing.JFrame padre) throws
    java.io.FileNotFoundException, java.io.IOException {
        selectorDeArchivo = new javax.swing.JFileChooser();
        selectorDeArchivo.setAcceptAllFileFilterUsed(false);
        selectorDeArchivo.addChoosableFileFilter(new TextFileFilter());
        seleccion = selectorDeArchivo.showOpenDialog(padre);
        if (seleccion == javax.swing.JFileChooser.APPROVE_OPTION){
            archivo = selectorDeArchivo.getSelectedFile();
            Abrir = new java.io.BufferedReader(new java.io.FileReader(archivo));
            java.lang.StringBuffer lineaTotal = new java.lang.StringBuffer("");
            String linea = Abrir.readLine();
            while (linea != null){
                lineaTotal.append(linea);
                lineaTotal.append(System.getProperty("line.separator"));
                linea = Abrir.readLine();
            }
            Abrir.close();
            return lineaTotal.toString();
        }
        return "";
    }

    public void guardarArchivoTXT(javax.swing.JFrame padre, String fuente) throws
    java.io.FileNotFoundException{
        selectorDeArchivo = new javax.swing.JFileChooser();
        selectorDeArchivo.setAcceptAllFileFilterUsed(false);
        selectorDeArchivo.addChoosableFileFilter(new TextFileFilter());
        seleccion = selectorDeArchivo.showSaveDialog(padre);
        if (seleccion == javax.swing.JFileChooser.APPROVE_OPTION){
            archivo =
            new
```

```

java.io.File(selectorDeArchivo.getSelectedFile().toString()+".txt");
    guardar = new java.io.PrintWriter(archivo);
    guardar.println(fuente);
    guardar.close();
}
}
private class TextFileFilter extends javax.swing.filechooser.FileFilter {
    public boolean accept(java.io.File f) {
        if (f.isDirectory())
            return true;
        String s = f.getName();
        int i = s.lastIndexOf('.');

        if (i > 0 && i < s.length() - 1)
            if (s.substring(i + 1).toLowerCase().equals("txt"))
                return true;
        return false;
    }
    public String getDescription() {
        return "Archivos de texto (*.txt)";
    }
}
}

```


Anexo S. Código de Servidor

```
package giobot.servidor;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class Servidor extends javax.swing.JFrame implements
gnu.io.SerialPortEventListener{
    public Servidor() {
        initComponents();
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                cerrarConexionTCPIP();
                System.exit(0);
            }
        });
    }
    private void mostrarDialogoAcercaDe(){
        SAcercaDe acerca = new SAcercaDe(this);
        acerca.setLocationRelativeTo(this);
        acerca.setVisible(true);
    }
    public void cerrarConexionTCPIP(){
        establecerCampoTextoEditable(false);
        if (servidor != null){
            if (servidor.isClosed() == false){
                try {
                    if (servidorVideo != null){
                        mostrarMensaje("\nCerrando captura de Vídeo...");
                        servidorVideo.stop();
                    }
                    if (prtoSerie != null){
                        mostrarMensaje("\nCerrando puerto serie...");
                        prtoSerie.cerrarPuertoSerie();
                    }
                    mostrarMensaje("\nFinalizando la conexión Cliente-Servidor...");
                    salida.close();
                    entrada.close();
                    conexion.close();
                } catch (java.io.IOException e){
                    mostrarMensaje("\nError - " + e.getMessage());
                } catch (NullPointerException e){
                    mostrarMensaje("\nError - " + e.getMessage());
                }
            }
        }
    }
}
```

```

    }
    }
    }
    private void conectarPuertoSerie() throws
gnu.io.UnsupportedCommOperationException,

gnu.io.PortInUseException,gnu.io.NoSuchPortException,java.util.TooManyListener
sException{
    enviaDatosDesdeServidorACliente("\nConectando al puerto serie...");
    prtoSerie = new puertoSerie();

    prtoSerie.conectarPuertoSerie(puertoSerial,baudios,paridad,bitDeParada,bitDeDat
os,controlDeFlujo);
    enviaDatosDesdeServidorACliente(prtoSerie.parametrosDeConexion());
    prtoSerie.start();
    prtoSerie.addEventListener(this);
}
private void mostrarDialogoPropiedades(){
    SPropiedades dialogo = new SPropiedades(this);
    dialogo.setLocationRelativeTo(this);
    dialogo.setVisible(true);
    hayOk = dialogo.leerOk();
    if(hayOk){
        hayVideoDisponible = dialogo.leerVideoDisponible();
        hayPuertoSerieDisponible = dialogo.leerPuertoSerieDisponible();
        puerto = dialogo.leerValorPuertoTCP();
        numeroDeConexiones = dialogo.leerNumeroDeConexiones();
        puertoVideo = dialogo.leerValorPuertoVideo();
        nombreDelDispositivoVideo = dialogo.leerNombreDelDispositivoDeVideo();
        puertoSerial = dialogo.leerPuertoSerie();
        baudios = dialogo.leerBaudios();
        paridad = dialogo.leerParidad();
        bitDeParada = dialogo.leerBitDeParada();
        bitDeDatos = dialogo.leerBitDeDatos();
        controlDeFlujo = dialogo.leerControlDeFlujo();
        cerrarConexionTCPIP();
        ejecutarServidor app = new ejecutarServidor();
    }
    System.runFinalization();
}
public class ejecutarServidor implements Runnable{
    @SuppressWarnings("CallToThreadStartDuringObjectConstruction")
    ejecutarServidor() {

```

```

        Thread ejecutar = new Thread(this, "ejecutar");
        ejecutar.start();
    }
    public void run(){
        try {
            servidor = new java.net.ServerSocket(puerto, numeroDeConexiones);
            while (true){
                try {
                    esperarConexionDeUnCliente();
                    obtenerFlujosDeEntradaSalida();
                    if (hayPuertoSerieDisponible) conectarPuertoSerie();
                    if (hayVideoDisponible) transmitirVideo();
                    procesarConexionTCPIP();
                } catch (java.io.EOFException eEOF){
                    mostrarMensaje("\nEl servidor terminó la conexión - " +
eEOF.getMessage());
                } catch (java.io.IOException e){
                    mostrarMensaje("\nError - " + e.getMessage());
                } finally {
                    cerrarConexionTCPIP();
                    ++contador;
                }
            }
        } catch (java.io.IOException e){
            mostrarMensaje("\nError - " + e.getMessage());
        } catch (gnu.io.UnsupportedCommOperationException e){
            mostrarMensaje("\nError - Operación del puerto serie NO soportada.");
        } catch (gnu.io.PortInUseException e) {
            mostrarMensaje("\nError - El puerto serie está siendo usado por otra
aplicación.");
        } catch (gnu.io.NoSuchPortException e) {
            mostrarMensaje("\nError - " + e.getMessage() + ". ");
        } catch (ClassNotFoundException e) {
            mostrarMensaje("\nError - Se recibió un tipo de objeto desconocido.");
        } catch (InterruptedException e) {
            mostrarMensaje("\nError - Ejecución interrumpida.");
        } catch (java.util.TooManyListenersException e) {
            mostrarMensaje("\nError - Muchos oyentes de eventos implementados.
");
        }
    }
}

public void enviaDatosDesdeServidorACliente(String mensaje){
    try {

```

```

        salida.writeObject("'" + mensaje);
        salida.flush();
        mostrarMensaje(mensaje);
    } catch (java.io.IOException e){
        mostrarMensaje("\nError - NO se escribió el objeto.");
    } catch (NullPointerException e) {
        mostrarMensaje("\nError - Es posible que NO se haya establecido una
conexión. " + e.getMessage());
    }
}

private void esperarConexionDeUnCliente() throws java.io.IOException{
    mostrarMensaje("\nEsperando una conexión...");
    conexion = servidor.accept();
    mostrarMensaje("\nConexión " + contador + " recibida de " +
conexion.getInetAddress().getHostName()+":
"+conexion.getInetAddress().getHostAddress());
}

private void establecerCampoTextoEditable( final boolean editable ){
    javax.swing.SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            campoIntroducir.setEditable(editable);
        }
    });
}

private void mostrarMensaje(final String mensajeAMostrar){
    javax.swing.SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            areaTexto.append(mensajeAMostrar);
            areaTexto.setCaretPosition(areaTexto.getText().length());
        }
    });
}

private void obtenerFlujosDeEntradaSalida() throws java.io.IOException{
    salida = new java.io.ObjectOutputStream(conexion.getOutputStream());
    salida.flush();
    entrada = new java.io.ObjectInputStream(conexion.getInputStream());
    mostrarMensaje("\nSe recibieron los flujos de E/S");
}

private void procesarConexionTCPIP() throws
java.io.IOException, InterruptedException, ClassNotFoundException{
    String mensaje = "\nConexión exitosa\n";
    enviaDatosDesdeServidorACliente(mensaje);
    establecerCampoTextoEditable(true);
    do {

```

```

        mensaje = (String) entrada.readObject();
        mostrarMensaje("Usuario>" + mensaje + "\n>");

        if (hayPuertoSerieDisponible && !mensaje.contains("@"))
            prtoSerie.enviaOrden(mensaje);
    } while (!mensaje.equals("DESCONECTAR"));
}

public void serialEvent(gnu.io.SerialPortEvent event) {
    try{
        leerPuertoSerie = prtoSerie.serialEvent(event);
        if (!leerPuertoSerie.equals("")){
            enviaDatosDesdeServidorACliente(leerPuertoSerie);
        }
    } catch (java.io.IOException e){
        mostrarMensaje("\nError - " + e.getMessage());
    } catch (java.lang.InterruptedExecution e){
        mostrarMensaje("\nError - " + e.getMessage());
    }
}

private void transmitirVideo(){
    enviaDatosDesdeServidorACliente("\nConectando la cámara...");
    camara =
    javax.media.CaptureDeviceManager.getDevice(nombreDelDispositivoVideo);
    servidorVideo = new
    STransmissionDeVideo(camara.getLocator(),conexion.getInetAddress().getHostAdd
    ress().toString(), puertoVideo);
    servidorVideo.start();
}
@SuppressWarnings("unchecked")

private void initComponents() {

    campoIntroducir = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    areaTexto = new javax.swing.JTextArea();
    barra = new javax.swing.JMenuBar();
    archivo = new javax.swing.JMenu();
    propiedades = new javax.swing.JMenuItem();
    jSeparator1 = new javax.swing.JPopupMenu.Separator();
    salir = new javax.swing.JMenuItem();
    ayuda = new javax.swing.JMenu();
    acercaDe = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

setTitle("Giobot Servidor 1.0.0");

setIconImage(java.awt.Toolkit.getDefaultToolkit().getImage("src/Giobot/Imagenes/
Giobot.png"));
setMinimumSize(new java.awt.Dimension(473, 300));
setResizable(false);

campoIntroducir.setFont(new java.awt.Font("Arial", 0, 12));
campoIntroducir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        campoIntroducirActionPerformed(evt);
    }
});

areaTexto.setBackground(new java.awt.Color(0, 0, 0));
areaTexto.setColumns(20);
areaTexto.setFont(new java.awt.Font("Courier New", 0, 12));
areaTexto.setForeground(new java.awt.Color(255, 255, 0));
areaTexto.setRows(5);
jScrollPane1.setViewportView(areaTexto);

archivo.setText("Archivo");
archivo.setFont(new java.awt.Font("Arial", 1, 12));

propiedades.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_P, java.awt.event.InputEvent.CTRL_MASK));
propiedades.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/propiedades.png")));
propiedades.setText("Propiedades...");
propiedades.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        propiedadesActionPerformed(evt);
    }
});
archivo.add(propiedades);
archivo.add(jSeparator1);

salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_F4, java.awt.event.InputEvent.ALT_MASK));
salir.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/salida.png")));

```

```

        salir.setText("Salir");
        salir.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                salirActionPerformed(evt);
            }
        });
        archivo.add(salir);

        barra.add(archivo);

        ayuda.setText("Ayuda");
        ayuda.setFont(new java.awt.Font("Arial", 1, 12));

        acercaDe.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_Y, java.awt.event.InputEvent.CTRL_MASK));
        acercaDe.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/giobot/Imagenes/Giobot.png")));
        acercaDe.setText("Acerca de...");
        acercaDe.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                acercaDeActionPerformed(evt);
            }
        });
        ayuda.add(acercaDe);
        barra.add(ayuda);
        setJMenuBar(barra);

        javax.swing.GroupLayout layout = new
        javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                    layout.createSequentialGroup()
                        .addContainerGap()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                            .add(ayuda)
                            .add(acercaDe)
                            .add(salir))
                        .addContainerGap()
                    )
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .add(salir)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .add(acercaDe)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .add(ayuda)
                    .addContainerGap()
                )
        );
        ventana.add(getContentPane());
        ventana.pack();
        ventana.setVisible(true);
    }

    private void salirActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void acercaDeActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
}

```

```

javax.swing.GroupLayout.DEFAULT_SIZE, 453, Short.MAX_VALUE))
    .addContainerGap()
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(campoIntroducir,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.DEFAULT_SIZE, 230, Short.MAX_VALUE)
        .addContainerGap()
    );
pack();
}

private void propiedadesActionPerformed(java.awt.event.ActionEvent evt) {
    mostrarDialogoPropiedades();
}

private void acercaDeActionPerformed(java.awt.event.ActionEvent evt) {
    mostrarDialogoAcercaDe();
}

private void campoIntroducirActionPerformed(java.awt.event.ActionEvent evt) {
    enviaDatosDesdeServidorACliente("\nUsuario<
"+evt.getActionCommand()+"\n");
    campoIntroducir.setText("");
}

private void salirActionPerformed(java.awt.event.ActionEvent evt) {
    cerrarConexionTCPIP();
    System.exit(0);
}

public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Windows".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
            }
        }
    }
}

```



```

        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Servidor.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            Servidor aplicacion = new Servidor();
            aplicacion.setVisible(true);

aplicacion.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
            aplicacion.setLocationRelativeTo(null);
        }
    });
}
private javax.swing.JMenuItem acercaDe;
private javax.swing.JMenu archivo;
private javax.swing.JTextArea areaTexto;
private javax.swing.JMenu ayuda;
private javax.swing.JMenuBar barra;
private javax.swing.JTextField campoIntroducir;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JPopupMenu.Separator jSeparator1;
private javax.swing.JMenuItem propiedades;
private javax.swing.JMenuItem salir;
private java.io.ObjectOutputStream salida;
private java.io.ObjectInputStream entrada;
private java.net.ServerSocket servidor;
private java.net.Socket conexion;

```

```
    private int contador = 1, baudios, bitDeDatos, puerto, numeroDeConexiones,  
    puertoVideo;  
    private String puertoSerial,paridad,bitDeParada,controlDeFlujo, leerPuertoSerie,  
    nombreDelDispositivoVideo;  
    private boolean hayOk,hayVideoDisponible,hayPuertoSerieDisponible;  
    private javax.media.CaptureDeviceInfo camara;  
    public puertoSerie prtoSerie;  
    public STransmisionDeVideo servidorVideo;  
}
```

Anexo T. Código de puertoSerie

```
package giobot.servidor;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class puertoSerie extends Thread {
    private String result, paridad, controlDeFlujo, bitDeParada;
    private int spParidad, spBitParada, spControl, numBytes;
    static java.util Enumeration portList;
    static gnu.io.CommPortIdentifier portId;
    static gnu.io.SerialPort puertoSerie;
    private java.io.BufferedInputStream entrada;
    private java.io.BufferedOutputStream salida;
    public puertoSerie() {}
    public void addEventListener(gnu.io.SerialPortEventListener oyente) throws
    java.util.TooManyListenersException{
        puertoSerie.addEventListener(oyente);
    }
    public void cerrarPuertoSerie() throws java.io.IOException{
        puertoSerie.close();
        salida.close();
        entrada.close();
    }
    public void conectarPuertoSerie(String Puerto, Integer Baudio, String Paridad,
    String BitParada, Integer BitDatos, String Control)
        throws
    gnu.io.UnsupportedCommOperationException,gnu.io.PortInUseException,gnu.io.No
    oSuchPortException {
        if (Paridad.equals("Ninguna")) spParidad =
    gnu.io.SerialPort.PARITY_NONE;
        if (Paridad.equals("Par")) spParidad = gnu.io.SerialPort.PARITY_EVEN;
        if (Paridad.equals("Impar")) spParidad = gnu.io.SerialPort.PARITY_ODD;
        if (Paridad.equals("Marcada")) spParidad =
    gnu.io.SerialPort.PARITY_MARK;
        if (Paridad.equals("Espaciada")) spParidad =
    gnu.io.SerialPort.PARITY_SPACE;
        if (BitParada.equals("1")) spBitParada = gnu.io.SerialPort.STOPBITS_1;
        if (BitParada.equals("1.5")) spBitParada =
    gnu.io.SerialPort.STOPBITS_1_5;
        if (BitParada.equals("2")) spBitParada = gnu.io.SerialPort.STOPBITS_2;
        if (Control.equals("Ninguno")) spControl =
    gnu.io.SerialPort.FLOWCONTROL_NONE;
```

```

        if (Control.equals("Xon/Xoff"))
            gnu.io.SerialPort.FLOWCONTROL_XONXOFF_IN
            gnu.io.SerialPort.FLOWCONTROL_XONXOFF_OUT;
        if (Control.equals("Hardware"))
            gnu.io.SerialPort.FLOWCONTROL_RTSCCTS_IN
            gnu.io.SerialPort.FLOWCONTROL_RTSCCTS_OUT;
        portList = gnu.io.CommPortIdentifier.getPortIdentifiers();
        while (portList.hasMoreElements()){
            portId = (gnu.io.CommPortIdentifier) portList.nextElement();
            if (portId.getPortType() == gnu.io.CommPortIdentifier.PORT_SERIAL) {
                if (portId.getName().equals(Puerto)){
                    puertoSerie = (gnu.io.SerialPort) portId.open("Giobot", 2000);
                    puertoSerie.setSerialPortParams(Baudio, BitDatos, spBitParada,
spParidad);
                    puertoSerie.setFlowControlMode(spControl);
                }
            }
        }
    }

    public void enviaOrden(String orden) throws
java.io.IOException, InterruptedException{
        if (portId!=null) {
            salida.write((orden+"\r").getBytes());
            salida.flush();
            Thread.sleep(2000);
        }
    }

    public String parametrosDeConexion(){
        return "\nPuerto: "+puertoSerie.getName()+"\nBaudios:
"+puertoSerie.getBaudRate()+"\nBit de datos: "+puertoSerie.getDataBits()+
"\nBit de parada: "+leerBitDeParada()+"\nParidad:
"+leerParidad()+"\nControl de Flujo: "+leerControlDeFlujo()+".";
    }

    private String leerParidad(){
        paridad="No se conectó al puerto.";
        if (puertoSerie.getParity() == 0) paridad = "Ninguna";
        if (puertoSerie.getParity() == 1) paridad = "Impar";
        if (puertoSerie.getParity() == 2) paridad = "Par";
        if (puertoSerie.getParity() == 3) paridad = "Marcada";
        if (puertoSerie.getParity() == 4) paridad = "Espaciada";
        return paridad;
    }

    private String leerControlDeFlujo(){
        controlDeFlujo="No se conectó al puerto.";
    }

```

```

        if (puertoSerie.getFlowControlMode() == 0) controlDeFlujo = "Ninguno";
        if (puertoSerie.getFlowControlMode() == 3) controlDeFlujo = "Hardware";
        if (puertoSerie.getFlowControlMode() == 12) controlDeFlujo = "XOn/XOff";
        return controlDeFlujo;
    }
    private String leerBitDeParada(){
        bitDeParada="No se conectó al puerto.";
        if (puertoSerie.getStopBits() == 1) bitDeParada = "1";
        if (puertoSerie.getStopBits() == 2) bitDeParada = "2";
        if (puertoSerie.getStopBits() == 3) bitDeParada = "1.5";
        return bitDeParada;
    }
    public void run() {
        super.run();
        try{
            salida = new
java.io.BufferedOutputStream(puertoSerie.getOutputStream());
            puertoSerie.notifyOnOutputEmpty(true);
            entrada = new java.io.BufferedInputStream(puertoSerie.getInputStream());
            puertoSerie.notifyOnDataAvailable(true);
            puertoSerie.notifyOnDSR(true);
        } catch (java.io.IOException e){ }
    }
    public String serialEvent(gnu.io.SerialPortEvent event) throws
java.io.IOException, java.lang.InterruptedException{
        switch (event.getEventType()) {
            case gnu.io.SerialPortEvent.BI:
            case gnu.io.SerialPortEvent.OE:
            case gnu.io.SerialPortEvent.FE:
            case gnu.io.SerialPortEvent.PE:
            case gnu.io.SerialPortEvent.CD:
            case gnu.io.SerialPortEvent.CTS:
            case gnu.io.SerialPortEvent.DSR:
                if (!puertoSerie.isDSR()) return "\nDSR: El dispositivo conectado al
puerto está apagado o fue desconectado el cable serial.";
                break;
            case gnu.io.SerialPortEvent.RI:
            case gnu.io.SerialPortEvent.OUTPUT_BUFFER_EMPTY:
                break;
            case gnu.io.SerialPortEvent.DATA_AVAILABLE:
                byte[] readBuffer = new byte[1024];
                while (entrada.available() > 0) {
                    numBytes = entrada.read(readBuffer);
                    Thread.sleep(20);
                }
            }
        }
    }

```

```
        result = new String(readBuffer,0,numBytes);  
        return result;  
    }  
    break;  
}  
return "";  
}  
}
```

Anexo U. Código de STransmisionDeVideo

```
package giobot.servidor;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class STransmisionDeVideo {
    private javax.media.MediaLocator localizador, localizadorSalidaVideo;
    private String direccionIP, rtpURLVideo, resultado;
    private int puertoVideo, ancho, alto;
    private boolean estadoVideo, hayPista = false, hayFalla = false;
    private Integer estadoBloqueado = new Integer(0);

    private javax.media.Processor procesador = null;
    private javax.media.DataSink transmisorRTPV = null;
    private javax.media.protocol.DataSource salidaDatosV = null, ds;
    private javax.media.protocol.ContentDescriptor descriptorDeContenido;
    private javax.media.Format formatosSoportados[], formatoElegido, jpegFmt,
h263Fmt;
    private javax.media.format.VideoFormat jpegFmto;
    private java.awt.Dimension tamano;
    private float frameRate;

    public STransmisionDeVideo(javax.media.MediaLocator localizadorMedia,
String dirIP, int puertoVideo) {
        this.localizador = localizadorMedia;
        this.direccionIP = dirIP;
        this.puertoVideo = puertoVideo;
    }
    public synchronized String start() {
        resultado = crearProcesador();
        if (resultado != null) return resultado;
        resultado = crearTransmisor();
        if (resultado != null) {
            procesador.close();
            procesador = null;
            return resultado;
        }
        procesador.start();
        return null;
    }
    public void stop() {
        synchronized (this) {
```

```

        if (procesador != null) {
            procesador.stop();
            procesador.close();
            procesador = null;
            transmisorRTPV.close();
            transmisorRTPV = null;
        }
    }
}

private String crearProcesador() {
    if (localizador == null) return "El Localizador de Vídeo es nulo.";
    try {
        ds = javax.media.Manager.createDataSource(localizador);
    } catch (Exception e) {
        return "No se pudo crear la Fuente de Vídeo";
    }
    try {
        procesador = javax.media.Manager.createProcessor(ds);
    } catch (javax.media.NoProcessorException e) {
        return "No se pudo crear el Procesador de Vídeo.";
    } catch (java.io.IOException e) {
        return "Error de E/S creando el Procesador de Vídeo.";
    }
    estadoVideo = esperarEstado(procesador,
    javax.media.Processor.Configured);
    if (estadoVideo == false) return "No se pudo configurar el Procesador de
    Vídeo.";

    javax.media.control.TrackControl [] pistas = procesador.getTrackControls();

    if (pistas == null || pistas.length < 1) return "No se encontró pistas en el
    Procesador de Vídeo.";

    descriptorDeContenido = new
    javax.media.protocol.ContentDescriptor(javax.media.protocol.ContentDescriptor.R
    AW_RTP);
    procesador.setContentDescriptor(descriptorDeContenido);
    hayPista = false;
    for (int i = 0; i < pistas.length; i++) {
        if (pistas[i].isEnabled()) {
            formatosSoportados = pistas[i].getSupportedFormats();
            if (formatosSoportados.length > 0) {
                if (formatosSoportados[0] instanceof
                javax.media.format.VideoFormat) {

```



```

        tamano =
        ((javax.media.format.VideoFormat)pistas[i].getFormat()).getSize();
        jpegFmt = new
        javax.media.Format(javax.media.format.VideoFormat.JPEG_RTP);
        h263Fmt = new
        javax.media.Format(javax.media.format.VideoFormat.H263_RTP);
        frameRate =
        ((javax.media.format.VideoFormat)pistas[i].getFormat()).getFrameRate();
        if (formatosSoportados[0].matches(jpegFmt)) {
            ancho = (tamano.width % 8 == 0 ? tamano.width :
            (int)(tamano.width / 8) * 8);
            alto = (tamano.height % 8 == 0 ? tamano.height :
            (int)(tamano.height / 8) * 8);
            formatoElegido = new
            javax.media.format.VideoFormat(javax.media.format.VideoFormat.JPEG_RTP,
            new java.awt.Dimension(ancho, alto), javax.media.Format.NOT_SPECIFIED,
            javax.media.Format.byteArray, frameRate);
        } else if (formatosSoportados[0].matches(h263Fmt)) {
            if (tamano.width < 128) {
                ancho = 128;
                alto = 96;
            } else if (tamano.width < 176) {
                ancho = 176;
                alto = 144;
            } else {
                ancho = 352;
                alto = 288;
            }
            formatoElegido = new
            javax.media.format.VideoFormat(javax.media.format.VideoFormat.H263_RTP,
            new java.awt.Dimension(ancho, alto), javax.media.Format.NOT_SPECIFIED,
            javax.media.Format.byteArray, frameRate);
        } else {
            formatoElegido = formatosSoportados[0];
        }
        } else formatoElegido = formatosSoportados[0];
        pistas[i].setFormat(formatoElegido);
        hayPista = true;
    } else pistas[i].setEnabled(false);
} else pistas[i].setEnabled(false);
}

if (!hayPista) return "No se pudo establecer alguna de las pistas de Vídeo
para un formato RTP válido.";

```

```

        estadoVideo = esperarEstado(procesador, javax.media.Controller.Realized);
        if (estadoVideo == false) return "No se pudo realizar el Procesador de
Vídeo.";

        escribirCalidadJPEG(procesador, 0.5f);

        salidaDatosV = procesador.getDataOutput();

        return null;
    }

    private String crearTransmisor() {
        rtpURLVideo = "rtp://" + direccionIP + ":" + puertoVideo + "/video";

        localizadorSalidaVideo = new javax.media.MediaLocator(rtpURLVideo);

        try {
            transmisorRTPV = javax.media.Manager.createDataSink(salidaDatosV,
localizadorSalidaVideo);
            transmisorRTPV.open();
            transmisorRTPV.start();
            salidaDatosV.start();
        } catch (javax.media.MediaException e) {
            return "No se pudo crear un RTP data sink";
        } catch (java.io.IOException e) {
            return "No se pudo crear un RTP data sink";
        }
        return null;
    }

    private void escribirCalidadJPEG(javax.media.Player p, float val) {

        javax.media.Control cs[] = p.getControls();
        javax.media.control.QualityControl qc = null;
        jpegFmto = new
javax.media.format.VideoFormat(javax.media.format.VideoFormat.JPEG);

        for (int i = 0; i < cs.length; i++) {

            if (cs[i] instanceof javax.media.control.QualityControl &&
                cs[i] instanceof javax.media.Owned) {
                Object owner = ((javax.media.Owned)cs[i]).getOwner();
                if (owner instanceof javax.media.Codec) {
                    javax.media.Format fmts[] =

```

```

((javax.media.Codec)owner).getSupportedOutputFormats(null);
    for (int j = 0; j < fmts.length; j++) {
        if (fmts[j].matches(jpegFmto)) {
            qc = (javax.media.control.QualityControl)cs[i];
            qc.setQuality(val);
            break;
        }
    }
    if (qc != null) break;
}
}
}
private Integer leerEstadoBloqueado() {
    return estadoBloqueado;
}
private void haFallado() {
    hayFalla = true;
}
private synchronized boolean esperarEstado(javax.media.Processor
procesadorDeFuncion, int estado) {
    procesadorDeFuncion.addControllerListener(new EstadoOyente());
    hayFalla = false;

    if (estado == javax.media.Processor.Configured) {
        procesadorDeFuncion.configure();
    } else if (estado == javax.media.Processor.Realized) {
        procesadorDeFuncion.realize();
    }

    while (procesadorDeFuncion.getState() < estado && !hayFalla) {
        synchronized (leerEstadoBloqueado()) {
            try {
                leerEstadoBloqueado().wait();
            } catch (InterruptedException ie) {
                return false;
            }
        }
    }

    if (hayFalla) return false;
    else return true;
}
class EstadoOyente implements javax.media.ControllerListener {

```

```

public void controllerUpdate(javax.media.ControllerEvent ce) {

    if (ce instanceof javax.media.ControllerClosedEvent)
        haFallado();

    if (ce instanceof javax.media.ControllerEvent) {
        synchronized (leerEstadoBloqueado()) {
            leerEstadoBloqueado().notifyAll();
        }
    }
}
}
}

```

Anexo V. Código de SPropiedades

```
package giobot.servidor;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class SPropiedades extends javax.swing.JDialog {

    public SPropiedades(java.awt.Frame parent) {
        super(parent, true);
        initComponents();
        setLocationRelativeTo(parent);
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                dispose();
            }
        });
        selectorPuertos.addItem("");
        selectorDispositivoDeVideo.addItem("");
        try {
            portList = gnu.io.CommPortIdentifier.getPortIdentifiers();
            while (portList.hasMoreElements()){
                portId = (gnu.io.CommPortIdentifier) portList.nextElement();
                if (portId.getPortType() == gnu.io.CommPortIdentifier.PORT_SERIAL) {
                    selectorPuertos.addItem(portId.getName());
                }
            }
            listaVideo = javax.media.CaptureDeviceManager.getDeviceList(new
            javax.media.format.VideoFormat(null));
            it = listaVideo.iterator();
            while (it.hasNext()){
                informacionDeCamara = (javax.media.CaptureDeviceInfo)it.next();
                nombreDelDispositivo = informacionDeCamara.getName();
                selectorDispositivoDeVideo.addItem(nombreDelDispositivo);
            }
        } catch (Exception e) {
            javax.swing.JOptionPane.showMessageDialog(null, "Error - " +
            e.getMessage(), "Giobot Servidor 1.0.0",
            javax.swing.JOptionPane.ERROR_MESSAGE);
        }
        @SuppressWarnings("unchecked")
        private void initComponents() {
```

```

botonAceptar = new javax.swing.JButton();
botonCancelar = new javax.swing.JButton();
tablaPadre = new javax.swing.JTabbedPane();
tablaConexion = new javax.swing.JPanel();
etiquetaPuerto = new javax.swing.JLabel();
etiquetaNumeroDeConexiones = new javax.swing.JLabel();
campoPuertoTCP = new javax.swing.JTextField();
campoNumeroDeConexiones = new javax.swing.JTextField();
botonRestablecer = new javax.swing.JButton();
etiquetaDispositivoDeVideo = new javax.swing.JLabel();
etiquetaPuertoVideo = new javax.swing.JLabel();
campoPuertoVideo = new javax.swing.JTextField();
selectorDispositivoDeVideo = new javax.swing.JComboBox();
botonActualizar = new javax.swing.JButton();
videoDisponible = new javax.swing.JCheckBox();
jSeparator1 = new javax.swing.JSeparator();
tablaSerie = new javax.swing.JPanel();
etiquetaBaudios = new javax.swing.JLabel();
etiquetaPuertoSerial = new javax.swing.JLabel();
etiquetaBitDeDatos = new javax.swing.JLabel();
etiquetaParidad = new javax.swing.JLabel();
etiquetaBitDeParada = new javax.swing.JLabel();
etiquetaControlDeFlujo = new javax.swing.JLabel();
selectorPuertos = new javax.swing.JComboBox();
selectorBaudios = new javax.swing.JComboBox();
selectorBitDeDatos = new javax.swing.JComboBox();
selectorParidad = new javax.swing.JComboBox();
selectorBitDeParada = new javax.swing.JComboBox();
selectorControlDeFlujo = new javax.swing.JComboBox();
puertoSerieDisponible = new javax.swing.JCheckBox();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Propiedades de Conexión");

setIconImage(java.awt.Toolkit.getDefaultToolkit().getImage("src/Giobot/Imagenes/Giobot.png"));
setModal(true);
setResizable(false);

botonAceptar.setFont(new java.awt.Font("Arial", 0, 12));
botonAceptar.setText("Aceptar");
botonAceptar.setPreferredSize(new java.awt.Dimension(83, 23));
botonAceptar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        botonAceptarActionPerformed(evt);
    }
});

botonCancelar.setFont(new java.awt.Font("Arial", 0, 12));
botonCancelar.setText("Cancelar");
botonCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        botonCancelarActionPerformed(evt);
    }
});

tablaPadre.setFont(new java.awt.Font("Arial", 1, 12));
tablaPadre.setOpaque(true);

etiquetaPuerto.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaPuerto.setText("Puerto :");

etiquetaNumeroDeConexiones.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaNumeroDeConexiones.setText("No. de Conexiones :");
etiquetaNumeroDeConexiones.setToolTipText("");

campoPuertoTCP.setFont(new java.awt.Font("Arial", 0, 12));
campoPuertoTCP.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
campoPuertoTCP.setText("60002");
campoPuertoTCP.setToolTipText("49152 – 65535");
campoPuertoTCP.setPreferredSize(new java.awt.Dimension(60, 20));
campoPuertoTCP.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        campoPuertoTCPActionPerformed(evt);
    }
});
campoPuertoTCP.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        campoPuertoTCPKeyTyped(evt);
    }
});

campoNumeroDeConexiones.setFont(new java.awt.Font("Arial", 0, 12));
campoNumeroDeConexiones.setHorizontalAlignment(javax.swing.JTextField.RIGH
HT);
campoNumeroDeConexiones.setText("100");
campoNumeroDeConexiones.setToolTipText("Cantidad de conexiones
soportadas por el Servidor.");

```

```

        campoNumeroDeConexiones.setMinimumSize(new java.awt.Dimension(60,
20));
        campoNumeroDeConexiones.setName("");
        campoNumeroDeConexiones.setPreferredSize(new java.awt.Dimension(60,
20));
        campoNumeroDeConexiones.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                campoNumeroDeConexionesActionPerformed(evt);
            }
        });
        campoNumeroDeConexiones.addKeyListener(new
java.awt.event.KeyAdapter() {
            public void keyTyped(java.awt.event.KeyEvent evt) {
                campoNumeroDeConexionesKeyTyped(evt);
            }
        });

        botonRestablecer.setFont(new java.awt.Font("Arial", 0, 12));
        botonRestablecer.setText("Restablecer");
        botonRestablecer.setToolTipText("");
        botonRestablecer.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                botonRestablecerActionPerformed(evt);
            }
        });

        etiquetaDispositivoDeVideo.setFont(new java.awt.Font("Arial", 1, 12));
        etiquetaDispositivoDeVideo.setText("Dispositivo Video :");
        etiquetaDispositivoDeVideo.setToolTipText("");
        etiquetaDispositivoDeVideo.setRequestFocusEnabled(false);

        etiquetaPuertoVideo.setFont(new java.awt.Font("Arial", 1, 12));
        etiquetaPuertoVideo.setText("Puerto Video :");
        etiquetaPuertoVideo.setToolTipText("");

        campoPuertoVideo.setFont(new java.awt.Font("Arial", 0, 12)); // NOI18N
        campoPuertoVideo.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
        campoPuertoVideo.setText("60004");
        campoPuertoVideo.setToolTipText("49152 – 65535");
        campoPuertoVideo.setPreferredSize(new java.awt.Dimension(60, 20));
        campoPuertoVideo.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                campoPuertoVideoActionPerformed(evt);
            }
        });

```



```

    }
    });
    campoPuertoVideo.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            campoPuertoVideoKeyTyped(evt);
        }
    });

    selectorDispositivoDeVideo.setFont(new java.awt.Font("Arial", 0, 12));
    selectorDispositivoDeVideo.setToolTipText("Dispositivo de Captura de Vídeo");
    selectorDispositivoDeVideo.addActionListener(new
    java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            selectorDispositivoDeVideoActionPerformed(evt);
        }
    });

    botonActualizar.setFont(new java.awt.Font("Arial", 0, 12));
    botonActualizar.setText("Actualizar");
    botonActualizar.setToolTipText("Busca si hay dispositivos de Captura de
    Vídeo disponibles.");
    botonActualizar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botonActualizarActionPerformed(evt);
        }
    });

    videoDisponible.setFont(new java.awt.Font("Arial", 1, 12));
    videoDisponible.setSelected(true);
    videoDisponible.setText("Video");
    videoDisponible.setToolTipText("Establece si desea trabajar con algun
    dispositivo de Captura de Vídeo.");
    videoDisponible.addChangeListener(new javax.swing.event.ChangeListener()
    {
        public void stateChanged(javax.swing.event.ChangeEvent evt) {
            videoDisponibleStateChanged(evt);
        }
    });

    javax.swing.GroupLayout tablaConexionLayout = new
    javax.swing.GroupLayout(tablaConexion);
    tablaConexion.setLayout(tablaConexionLayout);
    tablaConexionLayout.setHorizontalGroup(

```

```

tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(tablaConexionLayout.createSequentialGroup()

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(tablaConexionLayout.createSequentialGroup()
        .addGap(10, 10, 10)

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(etiquetaPuerto)
    .addComponent(etiquetaNumeroDeConexiones))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(campoNumeroDeConexiones,
        javax.swing.GroupLayout.Alignment.TRAILING,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(campoPuertoTCP,
        javax.swing.GroupLayout.Alignment.TRAILING,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        tablaConexionLayout.createSequentialGroup()
        .addContainerGap(33, Short.MAX_VALUE)
        .addComponent(botonActualizar)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(botonRestablecer))
    .addGroup(tablaConexionLayout.createSequentialGroup()
        .addGap(42, 42, 42)
        .addComponent(etiquetaPuertoVideo)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(campoPuertoVideo,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGroup(tablaConexionLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(videoDisponible)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jSeparator1,
javax.swing.GroupLayout.DEFAULT_SIZE, 162, Short.MAX_VALUE))
    .addGroup(tablaConexionLayout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(etiquetaDispositivoDeVideo)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(selectorDispositivoDeVideo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap()
);
tablaConexionLayout.setVerticalGroup(

tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(tablaConexionLayout.createSequentialGroup()
        .addContainerGap()

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(campoPuertoTCP,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(etiquetaPuerto))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(etiquetaNumeroDeConexiones)
    .addComponent(campoNumeroDeConexiones,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(17, 17, 17)

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(tablaConexionLayout.createSequentialGroup()
        .addComponent(jSeparator1,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(12, 12, 12))
        .addGroup(tablaConexionLayout.createSequentialGroup()
        .addComponent(videoDisponible)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
        .addComponent(selectorDispositivoDeVideo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(etiquetaDispositivoDeVideo))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
        .addComponent(etiquetaPuertoVideo)
        .addComponent(campoPuertoVideo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(tablaConexionLayout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.BASELINE)
        .addComponent(botonRestablecer)
        .addComponent(botonActualizar))
        .addContainerGap()
);

tablaPadre.addTab("Servidor", tablaConexion);

etiquetaBaudios.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaBaudios.setText("Baudios :");

etiquetaPuertoSerial.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaPuertoSerial.setText("Puerto :");

etiquetaBitDeDatos.setFont(new java.awt.Font("Arial", 1, 12));

```

```

etiquetaBitDeDatos.setText("Bit de datos :");

etiquetaParidad.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaParidad.setText("Paridad :");

etiquetaBitDeParada.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaBitDeParada.setText("Bit de parada :");

etiquetaControlDeFlujo.setFont(new java.awt.Font("Arial", 1, 12));
etiquetaControlDeFlujo.setText("Control :");

selectorPuertos.setFont(new java.awt.Font("Arial", 0, 12));
selectorPuertos.setMinimumSize(new java.awt.Dimension(94, 21));
selectorPuertos.setPreferredSize(new java.awt.Dimension(94, 21));

selectorBaudios.setFont(new java.awt.Font("Arial", 0, 12));
selectorBaudios.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "110", "300", "600", "1200", "2400", "4800", "9600", "19200", "38400",
"57600", "115200", "460800" }));
selectorBaudios.setSelectedIndex(7);
selectorBaudios.setToolTipText("");
selectorBaudios.setMinimumSize(new java.awt.Dimension(94, 21));
selectorBaudios.setPreferredSize(new java.awt.Dimension(94, 21));

selectorBitDeDatos.setFont(new java.awt.Font("Arial", 0, 12));
selectorBitDeDatos.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "5", "6", "7", "8" }));
selectorBitDeDatos.setSelectedIndex(3);
selectorBitDeDatos.setToolTipText("");
selectorBitDeDatos.setMinimumSize(new java.awt.Dimension(94, 21));
selectorBitDeDatos.setPreferredSize(new java.awt.Dimension(94, 21));

selectorParidad.setFont(new java.awt.Font("Arial", 0, 12));
selectorParidad.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Ninguna", "Par", "Impar", "Marcada", "Espaciada" }));
selectorParidad.setToolTipText("");
selectorParidad.setMinimumSize(new java.awt.Dimension(94, 21));
selectorParidad.setPreferredSize(new java.awt.Dimension(94, 21));

selectorBitDeParada.setFont(new java.awt.Font("Arial", 0, 12));
selectorBitDeParada.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "1", "1.5", "2" }));
selectorBitDeParada.setToolTipText("");
selectorBitDeParada.setMinimumSize(new java.awt.Dimension(94, 21));

```

```

selectorBitDeParada.setPreferredSize(new java.awt.Dimension(94, 21));

selectorControlDeFlujo.setFont(new java.awt.Font("Arial", 0, 12));
selectorControlDeFlujo.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Ninguno", "Xon/Xoff",
"Hardware" }));
selectorControlDeFlujo.setMinimumSize(new java.awt.Dimension(94, 21));
selectorControlDeFlujo.setPreferredSize(new java.awt.Dimension(94, 21));

puertoSerieDisponible.setFont(new java.awt.Font("Arial", 1, 12));
puertoSerieDisponible.setSelected(true);
puertoSerieDisponible.setText("Puerto Serie");
puertoSerieDisponible.setToolTipText("Establece si desea trabajar con el
puerto serie.");
puertoSerieDisponible.addChangeListener(new
javax.swing.event.ChangeListener() {
    public void stateChanged(javax.swing.event.ChangeEvent evt) {
        puertoSerieDisponibleStateChanged(evt);
    }
});

javax.swing.GroupLayout tablaSerieLayout = new
javax.swing.GroupLayout(tablaSerie);
tablaSerie.setLayout(tablaSerieLayout);
tablaSerieLayout.setHorizontalGroup(

tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(tablaSerieLayout.createSequentialGroup()
        .addContainerGap()

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(puertoSerieDisponible)
    .addGroup(tablaSerieLayout.createSequentialGroup()

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(etiquetaPuertoSerial)
    .addComponent(etiquetaBaudios)
    .addComponent(etiquetaBitDeDatos)
    .addComponent(etiquetaParidad)
    .addComponent(etiquetaBitDeParada)
    .addComponent(etiquetaControlDeFlujo))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING, false)
    .addComponent(selectorBitDeDatos,                                0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(selectorBaudios,
javax.swing.GroupLayout.Alignment.TRAILING, 0, 29, Short.MAX_VALUE)
    .addComponent(selectorBitDeParada,                                0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(selectorControlDeFlujo,                             0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(selectorParidad, 0, 94, Short.MAX_VALUE)
    .addComponent(selectorPuertos,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
    .addContainerGap(42, Short.MAX_VALUE))
);
tablaSerieLayout.setVerticalGroup(

tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(tablaSerieLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(puertoSerieDisponible)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
    .addComponent(etiquetaPuertoSerial)
    .addComponent(selectorPuertos,
javax.swing.GroupLayout.PREFERRED_SIZE,                             21,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)
    .addComponent(selectorBaudios,
javax.swing.GroupLayout.PREFERRED_SIZE,                             21,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addComponent(etiquetaBaudios,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
15,

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)
        .addComponent(selectorBitDeDatos,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(etiquetaBitDeDatos))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)
        .addComponent(selectorParidad,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(etiquetaParidad))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)
        .addComponent(selectorBitDeParada,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(etiquetaBitDeParada))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(tablaSerieLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)
        .addComponent(selectorControlDeFlujo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(etiquetaControlDeFlujo))
        .addContainerGap(14, Short.MAX_VALUE))

```



```

);

tablaPadre.addTab("Puerto Serie", tablaSerie);

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(botonAceptar,
                    javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(botonCancelar)
                    .addComponent(tablaPadre,
                        javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(10, 10, 10))
            .addContainerGap(10, true));
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(tablaPadre,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
            .addComponent(botonAceptar,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, true));
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(botonAceptar,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, true));
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(botonAceptar,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, true));

```

```

javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(botonCancelar))
    .addContainerGap()
);

tablaPadre.getAccessibleContext().setAccessibleName("Conexión");

pack();
} // </editor-fold>

private void botonAceptarActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        if (Integer.parseInt(campoPuertoTCP.getText()) >= 49152 &&
Integer.parseInt(campoPuertoTCP.getText()) <= 65535){
            puertoTCP = Integer.parseInt(campoPuertoTCP.getText());
            puertoTCPValido = true;
        }else{
            puertoTCPValido = false;
            getToolkit().beep();
            javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor para
el puerto en el intervalo de 49152 hasta 65535.", "Giobot Servidor 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
        }
        if (Integer.parseInt(campoNumeroDeConexiones.getText()) > 0){
            numeroDeConexiones =
Integer.parseInt(campoNumeroDeConexiones.getText());
            numeroDeConexionesValido = true;
        }else{
            numeroDeConexionesValido = false;
            getToolkit().beep();
            javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor
mayor que cero para la cantidad de conexiones.", "Giobot Servidor 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
        }
        if (videoDisponible.isSelected()){
            if (Integer.parseInt(campoPuertoVideo.getText()) >= 49152 &&
Integer.parseInt(campoPuertoVideo.getText()) <= 65535 &&
Integer.parseInt(campoPuertoVideo.getText()) % 2 == 0){
                puertoVideo = Integer.parseInt(campoPuertoVideo.getText());
                puertoVideoValido = true;
            }else{
                puertoVideoValido = false;
            }
        }
    }
}

```

```

        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor
PAR para el puerto de Video en el intervalo de 49152 hasta 65535.", "Giobot
Servidor 1.0.0", javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (!(selectorDispositivoDeVideo.getSelectedItem().toString().isEmpty() ||
selectorDispositivoDeVideo.getSelectedItem().toString().equals(""))){
        nombreDelDispositivoDeVideo =
selectorDispositivoDeVideo.getSelectedItem().toString();
        dispositivoDeVideoValido = true;
    }else{
        dispositivoDeVideoValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione un
dispositivo de captura de vídeo.", "Giobot Servidor 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (Integer.parseInt(campoPuertoVideo.getText()) !=
Integer.parseInt(campoPuertoTCP.getText()) &&
Integer.parseInt(campoPuertoTCP.getText()) !=
(Integer.parseInt(campoPuertoVideo.getText())+1)){
        sinInterferenciaEntrePuertosValido = true;
    }else{
        sinInterferenciaEntrePuertosValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor
diferente para Puerto que NO sea consecutivo del puerto para video.", "Giobot
Servidor 1.0.0", javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
} else {
    puertoVideoValido = true;
    dispositivoDeVideoValido = true;
    sinInterferenciaEntrePuertosValido = true;
}
if (puertoSerieDisponible.isSelected()){
    if (!(selectorPuertos.getSelectedItem().toString().isEmpty() ||
selectorPuertos.getSelectedItem().toString().equals(""))){
        puertoSerieSeleccionado =
selectorPuertos.getSelectedItem().toString();
        puertoSerieValido = true;
    }else{
        puertoSerieValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione un

```

```

puerto          serie.", "Giobot          Servidor          1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (Integer.parseInt(selectorBaudios.getSelectedItem().toString())>0){
        baudioSeleccionado          =
Integer.parseInt(selectorBaudios.getSelectedItem().toString());
        baudioValido = true;
    }else{
        baudioValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione la
velocidad          en          baudios.", "Giobot          Servidor          1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (Integer.parseInt(selectorBitDeDatos.getSelectedItem().toString())>0){
        bitDeDatosSeleccionado          =
Integer.parseInt(selectorBitDeDatos.getSelectedItem().toString());
        bitDeDatosValido = true;
    }else{
        bitDeDatosValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione un
tamaño          para          el          bit          de          datos.", "Giobot          Servidor          1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (!((String)selectorParidad.getSelectedItem()).isEmpty()){
        paridadSeleccionada = (String)selectorParidad.getSelectedItem();
        paridadValido = true;
    }else{
        paridadValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione la
paridad.", "Giobot          Servidor          1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (!((String)selectorBitDeParada.getSelectedItem()).isEmpty()){
        bitDeParadaSeleccionado          =
(String)selectorBitDeParada.getSelectedItem();
        bitDeParadaValido = true;
    }else{
        bitDeParadaValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione el bit
de          parada.", "Giobot          Servidor          1.0.0",

```

```

javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    if (!((String)selectorControlDeFlujo.getSelectedItem()).isEmpty()){
        controlDeFlujoSeleccionado =
        (String)selectorControlDeFlujo.getSelectedItem();
        controlDeFlujoValido = true;
    }else{
        controlDeFlujoValido = false;
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Seleccione el
control de flujo.", "Giobot Servidor 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
    } else {
        puertoSerieValido = true;
        baudioValido = true;
        bitDeDatosValido = true;
        paridadValido = true;
        bitDeParadaValido = true;
        controlDeFlujoValido = true;
    }
    if (sinInterferenciaEntrePuertosValido && dispositivoDeVideoValido &&
numeroDeConexionesValido &&
        puertoTCPValido && puertoVideoValido && puertoSerieValido &&
baudioValido && paridadValido &&
        bitDeDatosValido && bitDeParadaValido && controlDeFlujoValido){
        ok = true;
        dispose();
    } else ok = false;

    } catch (NumberFormatException e){
        getToolkit().beep();
        javax.swing.JOptionPane.showMessageDialog(null, "Dígite un valor para
los puertos en el intervalo de 49152 hasta 65535.", "Giobot Servidor 1.0.0",
javax.swing.JOptionPane.INFORMATION_MESSAGE);
    }
}

private void botonCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    ok = false;
    dispose();
}

private void campoPuertoVideoKeyTyped(java.awt.event.KeyEvent evt) {
    if (campoPuertoVideo.getText().length() >= 5 ) {
        getToolkit().beep();
    }
}

```

```

        evt.consume();
    }
    digitarSoloNumeros(evt);
}
private void campoNumeroDeConexionesKeyTyped(java.awt.event.KeyEvent
evt) {

    digitarSoloNumeros(evt);
}
private void campoPuertoTCPKeyTyped(java.awt.event.KeyEvent evt) {
    if (campoPuertoTCP.getText().length() >= 5 ) {
        getToolkit().beep();
        evt.consume();
    }
    digitarSoloNumeros(evt);
}
private void botonActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    selectorDispositivoDeVideo.removeAllItems();
    selectorDispositivoDeVideo.addItem("");
    listaVideo = javax.media.CaptureDeviceManager.getDeviceList(new
javax.media.format.VideoFormat(null));
    it = listaVideo.iterator();
    while (it.hasNext()){
        informacionDeCamara = (javax.media.CaptureDeviceInfo)it.next();
        nombreDelDispositivo = informacionDeCamara.getName();
        selectorDispositivoDeVideo.addItem(nombreDelDispositivo);
    }
}
private void botonRestablecerActionPerformed(java.awt.event.ActionEvent evt) {
    campoPuertoTCP.setText("60002");
    campoNumeroDeConexiones.setText("100");
    campoPuertoVideo.setText("60004");
}

private void campoNumeroDeConexionesActionPerformed(java.awt.event.ActionEvent evt) {
    campoNumeroDeConexiones.transferFocus();
}
private void selectorDispositivoDeVideoActionPerformed(java.awt.event.ActionEvent evt) {
    if (selectorDispositivoDeVideo.getSelectedItem() != null)

selectorDispositivoDeVideo.setToolTipText(selectorDispositivoDeVideo.getSelecte
dItem().toString());
}

```

```

    }
    private void videoDisponibleStateChanged(javax.swing.event.ChangeEvent evt)
    {
        selectorDispositivoDeVideo.setEnabled(videoDisponible.isSelected());
        campoPuertoVideo.setEnabled(videoDisponible.isSelected());
    }
    private void puertoSerieDisponibleStateChanged(javax.swing.event.ChangeEvent evt) {
        selectorPuertos.setEnabled(puertoSerieDisponible.isSelected());
        selectorBaudios.setEnabled(puertoSerieDisponible.isSelected());
        selectorBitDeDatos.setEnabled(puertoSerieDisponible.isSelected());
        selectorParidad.setEnabled(puertoSerieDisponible.isSelected());
        selectorBitDeParada.setEnabled(puertoSerieDisponible.isSelected());
        selectorControlDeFlujo.setEnabled(puertoSerieDisponible.isSelected());
    }

    private void campoPuertoVideoActionPerformed(java.awt.event.ActionEvent evt)
    {
        campoPuertoVideo.transferFocus();
        botonRestablecer.transferFocus();
    }
    private void campoPuertoTCPActionPerformed(java.awt.event.ActionEvent evt) {
        campoPuertoTCP.transferFocus();
    }
    private void digitarSoloNumeros(java.awt.event.KeyEvent evt){
        if(!(Character.isDigit(evt.getKeyChar()) || evt.getKeyChar() ==
java.awt.event.KeyEvent.VK_BACK_SPACE
        || evt.getKeyChar() == java.awt.event.KeyEvent.VK_DELETE ||
evt.getKeyChar() == java.awt.event.KeyEvent.VK_ENTER)) {
            getToolkit().beep();
            evt.consume();
        }
    }
    public int leerNumeroDeConexiones(){
        return numeroDeConexiones;
    }
    public int leerValorPuertoTCP(){
        return puertoTCP;
    }
    public String leerNombreDelDispositivoDeVideo(){
        return nombreDelDispositivoDeVideo;
    }
    public int leerValorPuertoVideo(){
        return puertoVideo;
    }

```

```

}
public boolean leerOk(){
    return ok;
}
public boolean leerVideoDisponible(){
    return videoDisponible.isSelected();
}
public boolean leerPuertoSerieDisponible(){
    return puertoSerieDisponible.isSelected();
}
public int leerBaudios(){
    return baudioSeleccionado;
}
public int leerBitDeDatos(){
    return bitDeDatosSeleccionado;
}
public String leerPuertoSerie(){
    return puertoSerieSeleccionado;
}
public String leerParidad(){
    return paridadSeleccionada;
}
public String leerBitDeParada(){
    return bitDeParadaSeleccionado;
}
public String leerControlDeFlujo(){
    return controlDeFlujoSeleccionado;
}
private javax.swing.JButton botonAceptar;
private javax.swing.JButton botonActualizar;
private javax.swing.JButton botonCancelar;
private javax.swing.JButton botonRestablecer;
private javax.swing.JTextField campoNumeroDeConexiones;
private javax.swing.JTextField campoPuertoTCP;
private javax.swing.JTextField campoPuertoVideo;
private javax.swing.JLabel etiquetaBaudios;
private javax.swing.JLabel etiquetaBitDeDatos;
private javax.swing.JLabel etiquetaBitDeParada;
private javax.swing.JLabel etiquetaControlDeFlujo;
private javax.swing.JLabel etiquetaDispositivoDeVideo;
private javax.swing.JLabel etiquetaNumeroDeConexiones;
private javax.swing.JLabel etiquetaParidad;
private javax.swing.JLabel etiquetaPuerto;
private javax.swing.JLabel etiquetaPuertoSerial;

```



```

private javax.swing.JLabel etiquetaPuertoVideo;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JCheckBox puertoSerieDisponible;
private javax.swing.JComboBox selectorBaudios;
private javax.swing.JComboBox selectorBitDeDatos;
private javax.swing.JComboBox selectorBitDeParada;
private javax.swing.JComboBox selectorControlDeFlujo;
private javax.swing.JComboBox selectorDispositivoDeVideo;
private javax.swing.JComboBox selectorParidad;
private javax.swing.JComboBox selectorPuertos;
private javax.swing.JPanel tablaConexion;
private javax.swing.JTabbedPane tablaPadre;
private javax.swing.JPanel tablaSerie;
private javax.swing.JCheckBox videoDisponible;
private boolean ok, sinInterferenciaEntrePuertosValido,
dispositivoDeVideoValido, numeroDeConexionesValido;
private boolean puertoTCPValido, puertoVideoValido, baudioValido,
bitDeDatosValido, puertoSerieValido, paridadValido, bitDeParadaValido,
controlDeFlujoValido;
private int puertoTCP=0, puertoVideo=0, numeroDeConexiones = 0,
baudioSeleccionado=0, bitDeDatosSeleccionado=0;
private String puertoSerieSeleccionado="",bitDeParadaSeleccionado="",
paridadSeleccionada="", controlDeFlujoSeleccionado="";
private String nombreDelDispositivo="",nombreDelDispositivoDeVideo="";
private java.util.Iterator it;
private java.util.Vector listaVideo=null;
static java.util Enumeration portList;
static gnu.io.CommPortIdentifier portId;
static gnu.io.SerialPort puertoSerie;
private javax.media.CaptureDeviceInfo informacionDeCamara;
}

```

Anexo W. Código de SAcercaDe

```
package giobot.servidor;
/**
 * @author Giovanni Zorrilla Prieto
 */
public class SAcercaDe extends javax.swing.JDialog {

    public SAcercaDe(java.awt.Frame parent) {
        super(parent, true);
        initComponents();
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                dispose();
            }
        });
    }
    @SuppressWarnings("unchecked")
    private void initComponents() {
        texto = new javax.swing.JPanel();
        autor = new javax.swing.JLabel();
        profesion = new javax.swing.JLabel();
        universidad = new javax.swing.JLabel();
        version = new javax.swing.JLabel();
        icono = new javax.swing.JLabel();
        cerrar = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Giobot Servidor 1.0.0");
        setBackground(java.awt.Color.black);
        setForeground(new java.awt.Color(204, 204, 204));
        setModal(true);
        setResizable(false);

        texto.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
        java.awt.Color(255, 204, 0)));

        autor.setFont(new java.awt.Font("Tahoma", 1, 11));
        autor.setText("Autor: Giovanni Zorrilla Prieto");

        profesion.setFont(new java.awt.Font("Tahoma", 1, 11));
        profesion.setText("Ingeniero Mecatrónico");
```

```

universidad.setFont(new java.awt.Font("Tahoma", 1, 11));
universidad.setText("Universidad Autónoma de Occidente");
universidad.setToolTipText("");

version.setFont(new java.awt.Font("Tahoma", 1, 11));
version.setText("Versión del producto: Giobot Servidor 1.0.0");

javax.swing.GroupLayout textoLayout = new javax.swing.GroupLayout(texto);
texto.setLayout(textoLayout);
textoLayout.setHorizontalGroup(

textoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(textoLayout.createSequentialGroup()
        .addGroup(textoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(textoLayout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(profesion, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
                .addGap(10, 10, 10)
                .addComponent(autor, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
                .addGap(10, 10, 10)
                .addComponent(universidad, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
            )
            .addGroup(textoLayout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(version, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
            )
        )
        .addGap(10, 10, 10)
    )
    .addGroup(textoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(textoLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(autor, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
            .addGap(10, 10, 10)
            .addComponent(universidad, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
            .addGap(10, 10, 10)
            .addComponent(version, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
        )
    )
);
textoLayout.setVerticalGroup(

textoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(textoLayout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(textoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(profesion, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
            .addComponent(autor, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
            .addComponent(universidad, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
        )
        .addGap(10, 10, 10)
        .addGroup(textoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(version, javax.swing.GroupLayout.DEFAULT_SIZE,
240, Short.MAX_VALUE)
        )
    )
);

```



```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(cerrar)
    .addContainerGap()
);

icono.getAccessibleContext().setAccessibleName("Icono");

pack();
}
private void cerrarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}
private javax.swing.JLabel autor;
private javax.swing.JButton cerrar;
private javax.swing.JLabel icono;
private javax.swing.JLabel profesion;
private javax.swing.JPanel texto;
private javax.swing.JLabel universidad;
private javax.swing.JLabel version;
}

```